

UNIVERSIDADE FEDERAL DO PARANÁ

MARIA JOSEANE FELIPE GUEDES MACÊDO

ALGORITMO DE PROCURA COM ESCOLHA DINÂMICA DAS
COORDENADAS PARA PROGRAMAÇÃO NÃO LINEAR COM
RESTRIÇÕES

CURITIBA

2017

MARIA JOSEANE FELIPE GUEDES MACÊDO

ALGORITMO DE PROCURA COM ESCOLHA DINÂMICA DAS
COORDENADAS PARA PROGRAMAÇÃO NÃO LINEAR COM
RESTRIÇÕES

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Matemática da Universidade Federal do Paraná para obtenção do título de Doutora em co-tutela de tese com a Universidade do Minho.

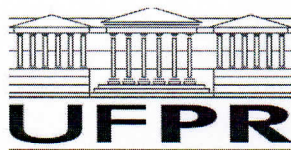
Orientadoras: Dra Elizabeth Wegner Karas

Dra M. Fernanda P. Costa

Dra Ana Maria A. C. Rocha

CURITIBA

2017



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor de Ciências Exatas
Programa de Pós-Graduação MATEMÁTICA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MATEMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **MARIA JOSEANE FELIPE GUEDES MACEDO** intitulada: **Algoritmo de procura com escolha dinâmica das coordenadas para programação não linear com restrições**, após terem inquirido a aluna e realizado a avaliação do trabalho, são de parecer pela sua **APROVAÇÃO** no rito de defesa. A outorga do título de doutor está sujeita à homologação pelo Colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 21 de Setembro de 2017.

ELIZABETH WEGNER KARAS
Presidente da Banca Examinadora

WELINGTON LUIS DE OLIVEIRA
Avaliador Externo

ANA ISABEL PINHEIRO NUNES PEREIRA
Avaliador Externo

GISLAÍNE APARECIDA PERICARO
Avaliador Externo

JOSÉ ALBERTO RAMOS FLOR
Avaliador Interno

Dedico este trabalho ao meu filho, Luis Henrique; aos meus pais, Guedes e Tiana; ao meu esposo, Álvaro; ao meu sobrinho, Pedro; e aos meus irmãos, Josicleide e Felipe.

Agradecimentos

Agradeço primeiramente a Deus, pois foi nele que me refugiei especialmente nas horas mais difíceis.

A minha família, em especial aos meus pais Guedes e Tiana, os melhores do mundo, que sempre se sacrificaram em prol da educação dos filhos. Pela maneira como conduziram nossa família, mesmo nos piores momentos nunca nos faltou nada, principalmente o amor. Vocês são o meu orgulho, a minha inspiração e têm participação especial em tudo isso. Aos meus irmãos Josicleide e Felipe, sempre presentes em minha vida mesmo que muitas vezes distantes fisicamente, amo muito vocês. Ao meu sobrinho Pedro, o amor de titia. Aos meus cunhados Gil e Wel. Obrigada a todos por cuidarem de Luis Henrique quando mais precisei.

Ao meu esposo Álvaro e ao meu filho Luis Henrique, pela compreensão e apoio incondicional que recebi de vocês, meus amores. A caminhada fica muito mais fácil com vocês ao meu lado, seja em Mossoró, em Curitiba ou em Portugal. Álvaro, meu amor, obrigada por tudo, essa tese é sua também. Não foram poucas as vezes que você abriu mão de se dedicar ao seu doutorado para que eu pudesse me dedicar ao meu. Luis Henrique, mamãe agradece todos os dias a Deus por você ser esse filho tão maravilhoso e compreensivo, apesar de tão novinho. Mamãe te ama além do infinito. Aos meus cunhados Rodrigo, Murilo e Manuela e ao meu sogro Luiz. Em especial a minha cunhada, Heloiza, por ter cuidado de Luis Henrique nos momentos que precisei me ausentar de Mossoró.

A minha orientadora, no Brasil, Dra. Elizabeth Karas, que desde o primeiro contato por email foi extremamente atenciosa. Por estar sempre presente, nessa etapa de minha vida, de modo que em momento algum me senti desamparada. Pelos ensinamentos de Matemática e de vida que me passou nesses anos de convívio. Espero que continuar com nossa parceria. Você é um grande exemplo na minha vida, muito obrigada por tudo.

As minhas orientadoras, no exterior, Dra. Maria Fernanda e Dra. Ana Rocha. Muito obrigada pela acolhida, na Universidade do Minho, pela atenção dedicada, pelo carinho, pelos ensinamentos e pela presença constante no decorrer deste trabalho. Foi um prazer

imenso trabalhar convosco e espero que continuemos com nossa parceria. Que sorte a minha, por ter tido a oportunidade de trabalhar com três orientadoras maravilhosas.

Aos membros da banca, em especial ao Dr. Welington de Oliveira, pela disponibilidade e pelas valiosas contribuições que permitiram o crescimento do trabalho.

À Universidade Federal do Paraná e ao Programa de Pós-graduação em Matemática (PPGM) na figura dos professoras, alunos e funcionários, muito obrigada pela ótima acolhida no curso de doutorado em Matemática. Aos amigos e parceiros de estudo que encontrei no decorrer dessa jornada, em especial Ana Chorobura, Monique, Adriano Delfino e Teles. Sofremos mas também demos muitas risadas juntos, sentirei saudades. A secretaria do PPGM, em especial à Cinthia, pela eficiência e presteza com que me auxiliou durante o doutorado.

À Universidade do Minho, pelo acolhimento durante o meu doutoramento sanduíche na Escola de Ciências e por aceitar o meu pedido de admissibilidade à preparação do Doutoramento em Ciências, especialidade em Matemática, em regime de cotutela.

À Universidade Federal Rural do Semi-Árido (UFERSA), da qual me orgulho em fazer parte, pela liberação para que pudesse me dedicar integralmente ao meu doutorado. Aos meus colegas de Matemática que me apoiaram disponibilizando-se a assumir disciplinas extras caso houvesse necessidade, em especial a Suene, Paulo César, Jackson e Elmer. A minha amiga Jusciane, que desviou sua rota na Europa para confraternizar o Natal conosco. Ao apoio financeiro da CAPES, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil, durante parte do doutorado no Brasil e pela bolsa de doutorado sanduíche no exterior.

A hospitalidade e o carinho da minha prima Adélia e das minhas amigas, Ju Carmona, Simone Bodanese, Adriana Belotto, Lucila Domingues, que muitas vezes me acolheram em suas casas em Curitiba.

A todas as pessoas que direta ou indiretamente passaram pela minha vida e de alguma forma contribuíram com a minha formação contínua.

Finalizo meus agradecimentos com o sentimento de que não é o fim de uma jornada, e sim o começo.

*“Não se cansem de trabalhar por um mundo mais justo
e solidário.”*

Papa Francisco

Resumo

Neste trabalho desenvolvemos um algoritmo geral estocástico de filtro, para resolver problemas de otimização não lineares e não convexos com restrições gerais. A generalidade deste algoritmo está no fato de que a análise de sua convergência quase certamente é garantida desde que a distribuição de probabilidade utilizada no cálculo dos iterandos satisfaça algumas hipóteses. O controle da inviabilidade é feito através da estratégia dos métodos de filtro. Baseados nesse algoritmo geral, desenvolvemos o Algoritmo FDDS, que baseia-se na ideia de busca com escolha dinâmica das coordenadas do Algoritmo DDS, para gerar os seus iterandos, e no método de filtro para controlar a inviabilidade. No FDDS os iterandos são calculados adicionando-se perturbações aleatórias com distribuição normal nas coordenadas, escolhidas de forma dinâmica, do melhor ponto corrente. No entanto, com a estratégia de gerar múltiplos pontos tentativos em cada iteração, o gasto com avaliações da função objetivo pode ser bastante elevado. Com o intuito de reduzir o número de avaliações de função, propomos o Algoritmo FDDSRBF, que também se encaixa na estrutura do algoritmo geral e cujos múltiplos pontos tentativos são gerados da mesma maneira que no FDDS. No entanto, o FDDSRBF utiliza um modelo cúbico de funções de base radial, para aproximar a função objetivo, na seleção do melhor ponto tentativo. Os algoritmos propostos não calculam ou aproximam quaisquer derivadas da função objetivo e das restrições. Resultados teóricos acerca das condições suficientes para a convergência quase certamente dos algoritmos foram apresentados. Resultados computacionais promissores, comparando-se o desempenho dos algoritmos propostos com alguns algoritmos existentes na literatura ao resolverem 42 problemas de três conjuntos diferentes, foram apresentados. O Algoritmo FDDSRBF mostrou-se bastante eficiente e robusto, com uma significativa redução do número de avaliações de função.

Palavras-chave: Métodos estocásticos; otimização global; algoritmo DDS; métodos de filtro.

Abstract

In this work we present an stochastic filter algorithm for solving nonlinear and nonconvex constrained global optimization problems. The generality of this algorithm lies in the fact that the analysis of its convergence is almost always guaranteed once the probability distribution used in the calculation of the iterates satisfies some hypotheses. The control of infeasibility is done through the strategy of the filter methods. Based on this general algorithm, we developed the FDDS algorithm, which combines the filter method with the dynamically dimensioned search algorithm. In the FDDS the iterates are calculated by adding random perturbations with normal distribution in the dynamically chosen coordinates of the best current point. However, with the strategy of generating multiple trial points in each iteration, the cost with objective function evaluations can be quite high. In order to reduce the number of function evaluations, we propose the FDDSRBF algorithm, which has the same general algorithm structure and whose multiple trial points are generated in the same way as in the FDDS. The FDDSRBF uses a cubic model of radial basis functions, to approximate the objective function, in the selection of the best trial point. The proposed algorithms do not compute or approximate any derivatives of the objective and constraint functions. Theoretical results concerning the sufficient conditions for the almost surely convergence of the proposed algorithms were presented. Promising computational results, in comparison to performance of the proposed algorithms with other algorithms in the literature when solving 42 problems of three different sets, were obtained. The FDDSRBF Algorithm provided competitive results when compared to the other methods.

Keywords: Stochastic methods; global optimization; DDS algorithm; filter methods.

Índice

Lista de Tabelas	xix
Lista de Figuras	xxii
Lista de algoritmos	xxiii
Notação e terminologia	xxvii
1 Introdução	1
1.1 Motivação	1
1.2 Contribuições da Tese	5
1.3 Estrutura da Tese	7
2 Conceitos da teoria de probabilidade	9
2.1 Espaços de probabilidade	9
2.1.1 Probabilidade condicional e independência	11
2.1.2 Variáveis aleatórias	12
2.1.3 Vetores aleatórios	15
2.1.4 Sequências de eventos e o Lema de Borel-Cantelli	19
2.1.5 Distribuição normal multivariada	19
2.2 Convergência no contexto probabilístico	21
3 Algoritmos estocásticos com iterandos viáveis	23
3.1 O problema	23

3.2	Um algoritmo geral	24
3.2.1	Algoritmo	24
3.2.2	Condições suficientes para convergência do algoritmo	26
3.3	Exemplos de aplicação da teoria de convergência	34
3.3.1	Algoritmo de busca aleatória localizada	34
3.3.2	O Algoritmo de Busca com Escolha Dinâmica das Coordenadas	38
3.3.3	O Algoritmo de Busca com Escolha Dinâmica das Coordenadas usando Modelos de Superfície de Resposta	44
4	Algoritmos estocásticos com controle de inviabilidade	53
4.1	O problema	54
4.2	Métodos de filtro	55
4.3	Um algoritmo geral estocástico de filtro	59
4.3.1	Algoritmo geral	59
4.3.2	Convergência do algoritmo	64
4.4	O Algoritmo de busca com escolha dinâmica das coordenadas e de filtro	69
4.4.1	Algoritmo FDDS	72
4.4.2	Convergência em probabilidade	74
4.5	Algoritmo FDDS usando modelos de aproximação por funções de base radial	75
4.5.1	Algoritmo FDDSRBF	79
4.5.2	Convergência em probabilidade	80
5	Experimentos numéricos	81
5.1	Detalhes de implementação	82
5.2	Resultados numéricos	83
5.2.1	Experimento para os problemas descritos em [3]	85
5.2.2	Experimento para os problemas descritos em [20]	91
5.2.3	Experimento para os problemas descritos em [47]	94
5.3	Análise do desempenho dos algoritmos	97

6	Conclusões e trabalhos futuros	99
	Referências	101

Lista de Tabelas

5.1	Resultados numéricos para os problemas de [3]	88
5.2	Resultados numéricos para os problemas de [20]	93
5.3	Resultados numéricos para os problemas de engenharia descritos em [47] .	96

Lista de Figuras

4.1	Regra de dominância de Pareto	56
4.2	Diferença entre uma região proibida com filtro reto e inclinado	57
4.3	Filtros reto e inclinado	58
4.4	Regiões proibidas no plano $f \times h$ para o filtro reto e inclinado	59
4.5	Caso em que X_k é viável	61
4.6	Critério de seleção do melhor ponto corrente	62
4.7	Seleção do melhor ponto tentativo no FDDS	71
4.8	Seleção do melhor ponto tentativo no FDDSRBF	78
5.1	Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da melhor rodada para os problemas descritos em [3]	86
5.2	Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da mediana das 30 rodadas para os problemas descritos em [3]	87
5.3	Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da melhor rodada para os problemas descritos em [20]	91
5.4	Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da média das 30 rodadas para os problemas descritos em [20]	92
5.5	Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da melhor rodada para os problemas descritos em [47]	95
5.6	Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da mediana das 30 rodadas para os problemas descritos em [47]	96

5.7	Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da melhor rodada para as três coleções de problemas teste	98
-----	--	----

Lista de algoritmos

A seguir listamos os algoritmos apresentados no trabalho:

- Algoritmo geral 25
- Algoritmo de busca aleatória localizada (LRS) 36
- Algoritmo de Busca com Escolha Dinâmica das Coordenadas (DDS) 41
- Algoritmo de Busca com Escolha Dinâmica das Coordenadas usando Modelos de Superfície de Resposta (DYCORS) 49
- Algoritmo geral estocástico de Filtro 60
- Algoritmo de Busca com Escolha Dinâmica das Coordenadas e de Filtro (FDDS) 72
- Algoritmo de Busca com Escolha Dinâmica das Coordenadas e de Filtro usando interpolação por Funções de Base Radial (FDDSRBF) 79

Notação e terminologia

A seguir listamos as notações e símbolos mais utilizados no trabalho:

- \mathbb{R} : conjunto dos números reais
- \mathbb{R}^n : espaço euclidiano n -dimensional
- $|\cdot|$: valor absoluto
- $\|\cdot\|$: norma vetorial ou matricial
- $B(z, \delta)$: bola aberta de centro em z e raio δ
- A^T : transposta da matriz A
- $\det(A)$: determinante da matriz quadrada A
- $\#I$: cardinal do conjunto I
- N^c : complementar do conjunto N
- Ω : espaço amostral abstrato
- \mathcal{F} : σ -álgebra de subconjuntos (eventos) aos quais se atribuirá uma probabilidade
- P : função que atribui probabilidades aos eventos
- $P(A)$: probabilidade do evento A
- $P(A|B)$: probabilidade condicional de A dado B

- (Ω, \mathcal{F}, P) : espaço de probabilidade
- (Ω, \mathcal{F}) : espaço mensurável
- μ : medida
- F_X : função de distribuição da variável aleatória X
- g : função de densidade
- $\mathcal{B}(\mathbb{R})$: subconjuntos de Borel em \mathbb{R}
- $\mathcal{B}(\Omega)$: σ -álgebra de Borel
- $\lim_{k \rightarrow \infty} A_k = [A_k \text{ infinitas vezes}]$: evento “ocorrência de um número infinito dos A_k ”
- Δ_I : operador diferença no intervalo I
- $U[0, 1]$: distribuição uniforme entre 0 e 1
- $N(\gamma, \lambda^2)$: distribuição normal com média γ e variância λ^2
- O símbolo “ \sim ” significa “tem como distribuição” ou “está distribuído como” (exemplo: $X \sim N(0, 1)$)
- $X_k \xrightarrow{P} X^*$: X_k converge em probabilidade para X^*
- $\{\Lambda_{k_j}\}$: coleção dos elementos aleatórios gerados na iteração k
- \mathcal{O}_k : oráculo associado à coleção de elementos aleatórios gerados até a iteração k
- Θ : função determinística dos elementos aleatórios de \mathcal{O}_k
- P_k : probabilidade da iteração k associada à escolha das coordenadas para perturbação aleatória
- h : função medida de inviabilidade
- F_k : filtro permanente da iteração k

- \bar{F}_k : filtro temporário da iteração k
- \mathcal{F}_k : região, na iteração k , permanentemente proibida em \mathbb{R}^n
- $\bar{\mathcal{F}}_k$: região, na iteração k , temporariamente proibida em \mathbb{R}^n
- \mathcal{A}_k : conjunto de pontos utilizados para alimentar o modelo RBF na iteração k
- S : função utilizada para aproximar a função objetivo na iteração k

Outras notações serão, naturalmente, introduzidas ao longo do texto.

Capítulo 1

Introdução

Esta tese enquadra-se na área da Otimização Não Linear. O objetivo da tese é desenvolver novos algoritmos estocásticos de otimização global eficientes e robustos para resolver problemas de otimização não lineares com restrições, não convexos e não suaves, em que avaliações da função objetivo são de custo computacional elevado.

1.1 Motivação

Muitos problemas que surgem de uma vasta gama de aplicações reais do cotidiano em diversas áreas são modelados como problemas não convexos de otimização não lineares com restrições. Devido ao contexto em que surgem estes problemas, é natural que em algumas aplicações reais não seja possível calcular as derivadas, ou porque o custo de cálculo é elevado ou porque as funções não estão disponíveis. Assim, é de todo inconveniente utilizar métodos que recorram ao uso de derivadas para determinar uma solução de um problema desta natureza. Neste contexto, os métodos estocásticos são atrativos devido à sua simplicidade computacional e por não envolverem o uso de derivadas analíticas ou numéricas. Por sua vez, problemas de otimização envolvendo modelos estocásticos ocorrem frequentemente em diversas áreas como, por exemplo engenharias, ciências, finanças, telecomunicações e medicina (ver [25, 47, 49]). Este tipo de problema é extremamente difícil

de resolver, o que estimula o interesse na análise e desenvolvimento de novos algoritmos de resolução eficientes e robustos para obter a solução de tais problemas. Devido à presença de parâmetros aleatórios, a combinação de conceitos da teoria de otimização com a teoria de probabilidade e estatística são primordiais. Tais parâmetros aleatórios podem estar presentes no problema e/ou no próprio algoritmo de otimização. Em geral, os métodos estocásticos foram inicialmente desenvolvidos para resolver problemas de otimização sem restrições, ou com restrições de limites simples nas variáveis de decisão (também designadas por restrições de caixa), e posteriormente estendidos para resolver problemas de otimização com restrições gerais. É de salientar que, o problema com restrições gerais é mais difícil de resolver e o método de otimização terá que incorporar estratégias para controle da inviabilidade dos iterandos ao longo do processo iterativo.

Uma importante classe de métodos para resolver problemas de otimização não lineares com restrições são os *métodos de penalidade*, que buscam a solução substituindo o problema com restrições original por uma sequência de subproblemas, ou apenas num, usando funções de penalidade, os quais são resolvidos por métodos eficientes de otimização não linear sem restrições. Nestes métodos, o objetivo é penalizar iterandos que não sejam viáveis, ao mesmo tempo que minimizam a função de penalidade (ver [1, 2, 21, 10]).

Em geral as funções de penalidade são definidas adicionando à função objetivo termos de penalidade (um por restrição), que são positivos se as restrições são violadas, caso contrário são nulos. Os termos de penalidade são multiplicados por um parâmetro de penalidade positivo. Nos métodos de penalidade, ao fazer-se o parâmetro de penalidade tender para infinito, vai-se penalizando cada vez mais severamente a violação às restrições, forçando assim que o minimizador da função de penalidade se aproxime da região viável do problema original. Deste modo, obtém-se uma sequência de minimizadores da função de penalidade que converge para a solução ótima do problema original. Estes métodos de penalidade são conhecidos na literatura por métodos de penalidade exterior. Existem na literatura diferentes termos de penalidade, tais como, linear, quadrático, estático, dinâmico, adaptativo, entre outros, e por conseguinte diferentes funções de penalidade, para mais detalhes ver [33, 34]. Uma questão crítica associada a tais métodos é a escolha apropriada

para o parâmetro de penalidade, o seu valor inicial e respectiva atualização ao longo do processo iterativo, de forma a manter o equilíbrio entre a função objetivo e os termos de penalidade.

Uma alternativa aos métodos de penalidade para resolver problemas de otimização com restrições são os *métodos de filtro* introduzidos por Fletcher e Leyffer em [15]. Os métodos de filtro baseiam-se no conceito de dominância, importado da otimização multi-objetivo, onde se constrói um filtro que aceita iterandos apenas se eles melhoram o valor da função objetivo ou o valor da função de medida da inviabilidade, e que é conhecida por *regra de dominância de Pareto*. Os métodos de filtro reformulam o problema de otimização com restrições num problema bi-objetivo em que minimizam a função objetivo e a função de medida da inviabilidade. Uma das vantagens dos métodos de filtro em relação aos métodos de penalidade é que não recorrem a funções de penalidade, ultrapassando-se assim a dificuldade existente com a atribuição do valor inicial, bem como os valores a atribuir durante o processo iterativo, ao parâmetro de penalidade. Estudos sobre a convergência global dos métodos de filtro podem ser encontrados em [6, 7, 14, 16, 17, 27, 35, 44].

Existem na literatura alguns algoritmos estocásticos baseados nos métodos de filtro. Por exemplo, em [46] é apresentado um método estocástico populacional baseado numa heurística que simula o comportamento de um cardume na água e no método de filtro. Neste método, o problema com restrições original é substituído por uma sequência de subproblemas bi-objetivos com restrições de limites simples nas variáveis. Cada subproblema é resolvido globalmente usando o algoritmo estocástico populacional cardume de peixes artificial e de filtro. Na resolução de cada subproblema, a população inicial é formada pelo minimizador do subproblema anterior e os pontos restantes são gerados aleatoriamente para explorar o espaço de busca por uma solução global do subproblema em resolução, e utiliza a metodologia de filtro para aceitar pontos não-dominados. Em [9] é apresentado um método estocástico de iniciação múltipla (do inglês, *multistart*) que incorpora como procedimento de busca local uma extensão do método de Hooke e Jeeves, ver [22], e de filtro para calcular múltiplas soluções de problemas de otimização não lineares com restrições e variáveis mistas, não convexos e não suaves.

Recentemente, surgiu na literatura um novo algoritmo estocástico designado por Algoritmo de Busca com Escolha Dinâmica das Coordenadas (do inglês, *Dynamically Dimensioned Search algorithm* - DDS), que é uma heurística estocástica desenvolvida por Tolson e Shoemaker [52], para resolver problemas de calibração automática de modelos de simulação de bacias hidrográficas. Dado que os problemas de calibração têm muitos parâmetros (variáveis) o algoritmo DDS apresenta-se como uma ferramenta simples e robusta para a resolução de tais problemas, os quais são demasiado dispendiosos em termos computacionais. Em [52] o desempenho do algoritmo DDS é comparado com o Algoritmo de Evolução Complexa Desordenada (do inglês, *shuffled complex evolution* - SCE) para um conjunto de problemas teste de otimização com restrições de limites simples nas variáveis e um problema modelo de formulações de calibração automática *SWAT2000* (do inglês, *Soil and Water Assessment Tool Documentation, version 2000* [32]). Os resultados apresentados mostraram que o DDS é mais eficiente do que o SCE, requerendo apenas entre 15% a 20% do número de avaliações da função para encontrar bons valores da função objetivo no conjunto de problemas testado. Além disso, também mostraram que o DDS converge rapidamente para boas soluções de calibração e evita facilmente ótimos locais.

Dada a eficácia do Algoritmo DDS na resolução de problemas complexos de calibração, têm surgido na literatura extensões do DDS bem como novas propostas de algoritmos estocásticos que incorporam a ideia do DDS. Por exemplo, em [51] é apresentada a extensão do DDS para resolver problemas de otimização não lineares com restrições de limites simples e envolvendo variáveis discretas, chamado de algoritmo de busca com escolha dinâmica das coordenadas híbrido. Em [39] é proposto o algoritmo ConstrLMSRBF, que usa modelos de superfície de funções de base radial (do inglês, *Radial Basis Function* - RBF), o qual é uma extensão do *Local Metric Stochastic RBF (LMSRBF) algorithm* proposto em [41] que incorpora a ideia do DDS. O algoritmo LMSRBF original foi desenvolvido para resolver problemas com restrições de limites simples nas variáveis com uma função objetivo do tipo caixa-preta de elevado custo computacional. O algoritmo ConstrLMSRBF é uma extensão do LMSRBF, para lidar com restrições de desigualdade do tipo caixa-preta e usa modelos de RBF para aproximar a função objetivo e as funções de restrição. Os resul-

tados computacionais reportados mostram que o algoritmo ConstrLMSRBF tem melhor desempenho do que os outros métodos de otimização usados para comparação.

Em [40] é proposto uma extensão do ConstrLMSRBF, designado por Extended ConstrLMSRBF, para resolver problemas do mesmo tipo como em [39], mas de grande dimensão. O Extended ConstrLMSRBF algoritmo está estruturado em duas fases. Na primeira fase, o algoritmo tenta encontrar um ponto viável. Uma vez encontrado o ponto viável, o algoritmo entra na segunda fase para melhorar esse ponto viável, procedendo de modo idêntico ao ConstrLMSRBF original. É de referir que em ambos os métodos, as restrições são tratadas individualmente através do uso de modelos de RBF, em vez do uso de funções de penalidade.

1.2 Contribuições da Tese

Nesta tese, tendo por base a eficácia demonstrada pelo DDS, a principal contribuição centra-se no desenvolvimento de um novo método estocástico que incorpora a ideia do DDS e a metodologia de filtro, para resolver problemas de otimização com restrições de igualdade e/ou de desigualdade, em que as avaliações da função objetivo são de custo computacional elevado.

Ao longo da tese considera-se o problema de otimização com restrições gerais, escrito na seguinte forma:

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeito a} && x \in \mathcal{D}_V, \end{aligned} \tag{1.1}$$

onde $f : A \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função determinística qualquer, x é o vetor que contém as variáveis de decisão e $\mathcal{D}_V \subseteq A \subseteq \mathbb{R}^n$ é o conjunto viável, sendo A o domínio de f . Estamos particularmente interessados nos casos em que avaliações da função objetivo são de custo computacional elevado, podendo esta ser do tipo caixa-preta. É de referir que, quando a função objetivo e/ou algumas das funções de restrição são não lineares o problema é classificado como sendo um problema de programação não linear (PNL). Mais ainda, o problema de PNL é classificado como não convexo quando a função objetivo é uma função

não convexa ou quando a região viável \mathcal{D}_V é um conjunto não convexo. É de salientar que, os problemas de PNL não convexos são considerados de difícil resolução dado que o problema poderá ter vários minimizadores locais e globais distintos. Se para além disto, alguma das funções envolvidas no problema é não diferenciável, então temos um problema de PNL não convexo e não suave, o que dificulta ainda mais a sua resolução.

O trabalho desenvolvido nesta tese centra-se na resolução de problemas de PNL com restrições, não convexos e não suaves, em que a função objetivo é de elevado custo computacional e não necessariamente contínua.

Esta tese está dividida em duas fases de estudo. Numa primeira fase são apresentados algoritmos estocásticos de otimização global, para a resolução de problemas de PNL com restrições de limites simples, não convexos e não suaves, entre eles o Algoritmo DDS. Para este tipo de problema, é fácil manter a viabilidade dos iterandos ao longo do processo iterativo. Os algoritmos apresentados geram os pontos iniciais dentro da região viável e usam estratégias de projeção dos iterandos na região viável, ao longo do processo iterativo para garantir a viabilidade. Nossa contribuição no contexto de problemas com restrições de limites simples na variáveis é o estudo da convergência em probabilidade do Algoritmo DDS, com base nas ideias de [38] e no Lema de Borel-Cantelli. Numa segunda fase, o estudo é direcionado para a resolução dos problemas de PNL com restrições gerais, não convexos e não suaves. Neste caso, a região viável \mathcal{D}_V é o conjunto definido por todos os pontos que satisfazem as restrições de desigualdade e de igualdade do problema, e onde algumas das funções, objetivo ou restrições, podem ser não lineares, não convexas e não suaves.

Neste contexto, propõe-se um algoritmo geral estocástico de filtro, onde os iterandos são vetores aleatórios cujas realizações são geradas de acordo com alguma distribuição de probabilidade e que usa uma metodologia de filtro para controlar a inviabilidade. Para o algoritmo geral estocástico de filtro, é efetuado um estudo das condições suficientes para que este seja convergente. Com base neste estudo do algoritmo geral estocástico de filtro, são propostos dois algoritmos estocásticos de filtro práticos que baseiam-se na ideia de busca com escolha dinâmica das coordenadas do DDS para gerar os seus iterandos. O

primeiro algoritmo prático proposto é denominado de *Algoritmo de Busca com Escolha Dinâmica das Coordenadas e de Filtro* (do inglês, *Filter based dynamically dimensioned search algorithm* - FDDS). E o segundo é denominado de *Algoritmo de Busca com Escolha Dinâmica das Coordenadas e de Filtro usando interpolação por Funções de Base Radial* (do inglês, *Filter-based Dynamically Dimensioned Search using Radial Basis Function interpolation* - FDDSRBF). É de salientar que o FDDSRBF proposto combina o FDDS com um modelo de interpolação da função objetivo por funções de base radial, para ultrapassar a dificuldade do elevado custo computacional associado às avaliações da função objetivo.

Para finalizar é de referir que, para avaliar os desempenhos dos algoritmos FDDS e FDDSRBF propostos, bem como a comparação com outros métodos de otimização, é efetuado um conjunto de experiências computacionais recorrendo a problemas teste disponíveis na literatura.

1.3 Estrutura da Tese

A tese está estruturada da seguinte forma. No Capítulo 2 apresenta-se uma revisão de alguns conceitos estatísticos básicos e resultados clássicos da teoria de medida em espaços de probabilidade. No Capítulo 3 é apresentado um estudo da convergência, no sentido probabilístico, de um algoritmo estocástico geral com iterandos viáveis para resolver problemas de otimização global com restrições de limites simples. No mesmo capítulo também são apresentados e descritos três algoritmos estocásticos de busca direta localizada que se encaixam nos moldes do algoritmo estocástico geral viável, entre eles o Algoritmo DDS. É também apresentado, um estudo das condições suficientes para que o Algoritmo DDS seja convergente em probabilidade. No Capítulo 4, o qual teve início durante o doutoramento em regime de cotutela realizado na Universidade do Minho, é proposto um algoritmo geral estocástico de filtro, teórico, e é apresentado um estudo da convergência em probabilidade deste algoritmo. Por fim, são desenvolvidos e propostos dois algoritmos práticos baseados no algoritmo teórico, denominados por FDDS e FDDSRBF. No Capítulo 5, apresentam-se as experiências computacionais realizadas com os algoritmos propostos, usando três conjun-

tos de problemas teste retirados da literatura. Para finalizar, no Capítulo 6 apresentam-se as conclusões deste trabalho e propostas de investigação para trabalho futuro.

Capítulo 2

Conceitos da teoria de probabilidade

Neste capítulo revisamos algumas definições e alguns resultados clássicos da teoria de medida em espaços de probabilidade necessários ao bom entendimento do texto. O objetivo aqui é fornecer uma introdução para os leitores que não estão familiarizados com tais conceitos e resultados. Se não for o caso, o leitor pode seguir para o Capítulo 3. Para uma leitura mais completa acerca do assunto, consultar [12, 23, 43, 48].

2.1 Espaços de probabilidade

Esta seção é baseada em [12, Sec. 1.1] e tem o objetivo de revisar os principais conceitos de espaços de probabilidade.

Um espaço amostral é definido como sendo o conjunto de todos os possíveis resultados de um experimento aleatório. Denotaremos um espaço amostral abstrato por Ω . Em geral, não é possível atribuir probabilidades a todos os subconjuntos de um dado conjunto. Um conjunto ao qual se atribui uma probabilidade é chamado de *evento aleatório*. Denotar-se-á por \mathcal{F} a classe dos conjuntos aos quais se atribuirá uma probabilidade. Como Ω representa o conjunto de todos os resultados possíveis, a probabilidade de Ω é igual 1. Portanto Ω deve pertencer a \mathcal{F} .

Definição 2.1 *Uma classe de conjuntos \mathcal{F} que satisfaz as seguintes propriedades*

(i) $\Omega \in \mathcal{F}$;

(ii) se $A \in \mathcal{F}$ então $A^c \in \mathcal{F}$; e

(iii) se $A_i \in \mathcal{F}$ é uma família enumerável de conjuntos então $\bigcup_i A_i \in \mathcal{F}$

é chamada de σ -álgebra.

Uma vez que $\bigcap_i A_i = \left(\bigcup_i A_i^c \right)^c$ segue que uma σ -álgebra é fechada sobre uma interseção enumerável.

Definição 2.2 Um espaço de probabilidade é uma tripla (Ω, \mathcal{F}, P) , onde Ω representa o espaço amostral correspondente aos resultados de um experimento, \mathcal{F} é uma σ -álgebra de subconjuntos (eventos) de Ω e $P : \mathcal{F} \rightarrow [0, 1]$ é uma função que atribui probabilidades aos eventos.

Ao omitirmos P , a dupla (Ω, \mathcal{F}) é chamada um espaço mensurável, isto é, um espaço no qual podemos introduzir uma medida.

Definição 2.3 Uma medida é uma função $\mu : \mathcal{F} \rightarrow \mathbb{R}$ que satisfaz as seguintes propriedades:

(i) $\mu(\emptyset) = 0$;

(ii) $\mu(A) \geq 0, \quad \forall A \in \mathcal{F}$;

(iii) Se $A_i \in \mathcal{F}$ é uma sequência enumerável de conjuntos disjuntos, então

$$\mu \left(\bigcup_i A_i \right) = \sum_i \mu(A_i). \quad (2.1)$$

Se $\mu(\Omega) = 1$ então μ é uma medida de probabilidade, que é usualmente denotada por P . Aqui a função $P : \mathcal{F} \rightarrow [0, 1]$ é uma medida de probabilidade, logo satisfaz a Definição 2.3 e $P(\Omega) = 1$.

As propriedades a seguir são consequências diretas da definição de uma medida de probabilidade P .

Propriedade 2.4 *Sejam A e B eventos num espaço de probabilidade (Ω, \mathcal{F}, P) .*

$$(a) \ P(A^c) = 1 - P(A);$$

$$(b) \ P(A \cup B) = P(A) + P(B) - P(A \cap B);$$

$$(c) \ (\text{Monotonicidade}) \ \text{Se } A \subseteq B \ \text{então } P(A) \leq P(B);$$

$$(d) \ (\text{Subaditividade}) \ \text{Para eventos } A_k, \text{ com } k \geq 1, \text{ tem-se}$$

$$P\left(\bigcup_{k=1}^{\infty} A_k\right) \leq \sum_{k=1}^{\infty} P(A_k); \quad (2.2)$$

$$(e) \ (\text{Continuidade}) \ A \ \text{medida } P \ \text{é contínua para sequências monótonas:}$$

$$(i) \ \text{Se } A_k \uparrow A \ (\text{isto é, } A_1 \subset A_2 \subset \dots \text{ e } \bigcup_k A_k = A) \ \text{então } P(A_k) \uparrow P(A).$$

$$(ii) \ \text{Se } A_k \downarrow A \ (\text{isto é, } A_1 \supset A_2 \supset \dots \text{ e } \bigcap_k A_k = A) \ \text{então } P(A_k) \downarrow P(A).$$

2.1.1 Probabilidade condicional e independência

Esta seção está baseada em [23, Sec. 1.2 e 1.3] e tem o objetivo de revisar conceitos e resultados sobre independência e probabilidade condicional.

A probabilidade condicional permite analisar o resultado de um evento quando existe alguma intervenção no espaço amostral.

Definição 2.5 *Seja (Ω, \mathcal{F}, P) um espaço de probabilidade. Se $B \in \mathcal{F}$ e $P(B) > 0$, a probabilidade condicional de A dado B , denotada por $P(A|B)$, é definida por*

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad A \in \mathcal{F}. \quad (2.3)$$

A expressão (2.3) também é conhecida como regra do produto.

A probabilidade condicional é uma medida de probabilidade e portanto satisfaz a Definição 2.3 e $P(\Omega|B) = 1$, bem como as Propriedades 2.4. Além disso, dada uma coleção de eventos $\{A_1, A_2, \dots, A_k\}$, é possível mostrar por indução que,

$$P(A_1 \cap A_2 \cap \dots \cap A_k) = P(A_1)P(A_2|A_1) \dots P(A_k|A_1 \cap A_2 \cap \dots \cap A_{k-1}), \quad (2.4)$$

para todo $A_1, A_2, \dots, A_k \in \mathcal{F}$ e para todo $k = 2, 3, \dots$

Quando dois eventos são independentes, a ocorrência de um não exerce nenhuma influência na probabilidade de ocorrência do outro.

Definição 2.6 *Seja (Ω, \mathcal{F}, P) um espaço de probabilidade. Os eventos aleatórios A e B são independentes se*

$$P(A \cap B) = P(A)P(B). \quad (2.5)$$

Definição 2.7 *(a) Os eventos A_1, A_2, \dots, A_k , com $k \geq 2$, são ditos (coletivamente) independentes se toda subfamília finita deles é de eventos independentes; isto é,*

$$P(A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_m}) = P(A_{i_1})P(A_{i_2}) \cdots P(A_{i_m}), \quad (2.6)$$

para todos $1 \leq i_1 \leq i_2 \leq \dots \leq i_m$ e $m = 2, 3, \dots, k$.

(b) Os eventos A_1, A_2, \dots são independentes se para todo $k \geq 2$, A_1, A_2, \dots, A_k são independentes.

(c) Seja I um conjunto de índices tal que $\#I \geq 2$. Os eventos A_i , com $i \in I$, são independentes se toda subfamília finita deles é de eventos independentes, isto é, se $A_{i_1}, A_{i_2}, \dots, A_{i_m}$ são independentes para toda combinação $\{i_1, i_2, \dots, i_m\}$ de elementos de I e todo $m = 2, 3, \dots$

2.1.2 Variáveis aleatórias

Esta seção está baseada em [23, Sec. 2.1-2.3] e tem o objetivo de revisar conceitos e resultados sobre variáveis aleatórias.

Quando o resultado de um experimento é um número real, o próprio resultado será o valor de uma variável aleatória, definida por $X(\omega) = \omega$. Por exemplo, considere escolher um ponto ao acaso no intervalo fechado $[0, 1]$ e seja X o valor do resultado, então $\Omega = [0, 1]$ e $X(\omega) = \omega$. Neste caso, cada variável aleatória é uma função real do resultado do experimento, onde X é a função identidade. Diremos que $X(\omega)$ é variável aleatória se, e somente se, o evento $[X \leq x] := \{\omega \in \Omega : X(\omega) \leq x\}$ for aleatório para todo $x \in \mathbb{R}$.

Definição 2.8 Uma variável aleatória X num espaço de probabilidade (Ω, \mathcal{F}, P) é uma função real definida no espaço Ω tal que $[X \leq x]$ é evento aleatório para todo $x \in \mathbb{R}$; isto é, $X : \Omega \rightarrow \mathbb{R}$ é variável aleatória se $[X \leq x] \in \mathcal{F}$ para todo $x \in \mathbb{R}$.

Um exemplo de uma variável aleatória é a *função indicadora* de um conjunto $A \in \mathcal{F}$ definida como segue.

$$1_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A. \end{cases} \quad (2.7)$$

Definição 2.9 A função de distribuição da variável aleatória X , denotada por F_X , é definida por

$$F_X(x) = P(X \leq x), \quad x \in \mathbb{R}. \quad (2.8)$$

Na literatura, a função de distribuição de X é frequentemente denominada de *função de distribuição acumulada* de X .

A seguir enunciamos algumas propriedades que a função de distribuição satisfaz.

Propriedade 2.10 Se X é uma variável aleatória então sua função de distribuição F_X satisfaz as seguintes propriedades:

- (a) Se $x \leq y$, então $F_X(x) \leq F_X(y)$. Isto é, F_X é não decrescente;
- (b) $\lim_{x \rightarrow \infty} F_X(x) = 1$ e $\lim_{x \rightarrow -\infty} F_X(x) = 0$;
- (c) Se $x_k \downarrow x$, então $F_X(x_k) \downarrow F_X(x)$. Isto é, F_X é contínua à direita.

Definição 2.11 Seja X uma variável aleatória num espaço de probabilidade (Ω, \mathcal{F}, P) .

- (a) Dizemos que X é discreta se existe um conjunto finito ou enumerável $\{x_1, x_2, \dots\} \subset \mathbb{R}$ tal que $X(\omega) \in \{x_1, x_2, \dots\}$, para todo $\omega \in \Omega$. A função $p(x_i)$ definida por $p(x_i) = P(X = x_i)$, com $i = 1, 2, \dots$, é chamada de função de probabilidade (ou função de frequência) de X .

(b) Dizemos que X é (absolutamente) contínua se existe uma função g mensurável, definida para todo x real, com $g(x) \geq 0$, tal que

$$F_X(x) = \int_{-\infty}^x g(t) dt, \quad \text{para todo } x \in \mathbb{R}. \quad (2.9)$$

Neste caso, dizemos que g é a função de densidade de probabilidade de X ou, simplesmente, densidade de X .

Uma função g é densidade de alguma variável aleatória X se, e somente se, $\int_{-\infty}^{\infty} g(x) dx = 1$.

Observação 2.12 (Conjuntos de Borel na reta) Suponha $\Omega = \mathbb{R}$ e seja

$$\mathcal{C} = \{(a, b], -\infty \leq a \leq b < \infty\}.$$

Defina os subconjuntos de Borel em \mathbb{R} por $\mathcal{B}(\mathbb{R}) := \sigma(\mathcal{C})$, onde $\sigma(\mathcal{C})$ é a σ -álgebra gerada por \mathcal{C} . Assim, os subconjuntos de Borel em \mathbb{R} são elementos da σ -álgebra gerada pelos intervalos que são abertos à esquerda e fechados à direita.

Proposição 2.13 [23, Prop. 2.1] Se X é uma variável aleatória em (Ω, \mathcal{F}, P) , então o evento

$$[X \in \mathcal{B}] := \{\omega \in \Omega : X(\omega) \in \mathcal{B}\} \quad (2.10)$$

é evento aleatório para todo Boreliano \mathcal{B} . Isto é,

$$[X \in \mathcal{B}] \in \mathcal{F}, \text{ para todo } \mathcal{B} \in \mathcal{B}(\Omega) := \sigma\text{-álgebra de Borel}, \quad (2.11)$$

onde $\mathcal{B}(\Omega)$ é a menor σ -álgebra contendo os intervalos.

Uma consequência da Proposição 2.13 é que as probabilidades $P(X \in \mathcal{B})$ são determinadas pela função de distribuição F_X . Além disso, a probabilidade definida na σ -álgebra de Borel, $P(X \in \mathcal{B})$, é chamada de *distribuição de X* . Existem várias representações da distribuição de uma variável aleatória X , onde geralmente é escolhida a representação mais conveniente. No caso contínuo, a distribuição de uma variável aleatória X pode ser representada tanto por meio da função de distribuição de probabilidade F_X como pela função de densidade g , porém é mais comum trabalhar com a função de densidade.

Proposição 2.14 [23, Proposição 2.2 (b)] *Se X é uma variável aleatória absolutamente contínua com densidade g , então*

$$P(X \in \mathcal{B}) = \int_{\mathcal{B}} g(x) dx, \quad (2.12)$$

para todo $\mathcal{B} \in \mathcal{B}(\Omega)$.

Exemplo 2.15 *A variável aleatória X possui distribuição normal padrão, denotada por $X \sim N(0, 1)$, se X tem densidade dada por*

$$g(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2), \quad x \in \mathbb{R}. \quad (2.13)$$

Considere $b > 0$ e $c \in \mathbb{R}$. Se X é uma variável aleatória, então $Y = c + bX$ também é uma variável aleatória. Pois $Y \leq y$ se, e somente se, $X \leq \frac{y-c}{b}$, de modo que o evento $[Y \leq y]$ é aleatório para todo $y \in \mathbb{R}$. A proposição a seguir mostra como obter a densidade de Y a partir da densidade de X .

Proposição 2.16 [23, Proposição 2.3] *Sejam X uma variável aleatória contínua com densidade g_X e $Y = c + bX$, onde $b > 0$ e $c \in \mathbb{R}$. Então Y tem densidade g_Y dada por,*

$$g_Y(y) = \frac{1}{b} g_X\left(\frac{y-c}{b}\right), \quad y \in \mathbb{R}. \quad (2.14)$$

Exemplo 2.17 *Se $X \sim N(0, 1)$ e $Y = \gamma + \lambda X$, com $\gamma > 0$ e $\lambda \in \mathbb{R}$, então da Proposição 2.16 a variável aleatória Y tem densidade dada por*

$$g(y) = \frac{1}{\lambda\sqrt{2\pi}} \exp\left(\frac{-(y-\gamma)^2}{2\lambda^2}\right), \quad y \in \mathbb{R}. \quad (2.15)$$

Neste caso $Y \sim N(\gamma, \lambda^2)$ e dizemos que Y tem distribuição normal com média γ e variância λ^2 .

2.1.3 Vetores aleatórios

Agora vamos nos basear em [23, Sec. 2.4] para estender nossa revisão sobre variáveis aleatórias para vetores aleatórios em \mathbb{R}^n .

Definição 2.18 Um vetor $\tilde{X} = (X_1, X_2, \dots, X_n)$, cujas componentes são variáveis aleatórias definidas no mesmo espaço de probabilidade (Ω, \mathcal{F}, P) , é chamado vetor aleatório (ou variável aleatória n -dimensional). Note que o vetor aleatório \tilde{X} é uma função de Ω em \mathbb{R}^n , isto é, $\tilde{X} : \Omega \rightarrow \mathbb{R}^n$.

Definição 2.19 A função de distribuição $F = F_{\tilde{X}} = F_{X_1, \dots, X_n}$ de um vetor aleatório $\tilde{X} = (X_1, \dots, X_n)$ é definida da seguinte maneira,

$$F(\tilde{x}) = F(x_1, \dots, x_n) = P(X_1 \leq x_1, \dots, X_n \leq x_n), \quad \forall (x_1, \dots, x_n) \in \mathbb{R}^n. \quad (2.16)$$

Além disso, F também é chamada de função de distribuição conjunta das variáveis aleatórias X_1, \dots, X_n .

O evento $[X_1 \leq x_1, \dots, X_n \leq x_n] := \bigcap_{i=1}^n [X_i \leq x_i]$ é aleatório, já que os componentes X_i são variáveis aleatórias e, portanto, $[X_i \leq x_i] \in \mathcal{F}$, para todo $i = 1, \dots, n$.

Veremos algumas propriedades da função de distribuição F de um vetor aleatório, mas antes precisamos da seguinte definição.

Definição 2.20 Sejam $I = (a, b]$ um intervalo e $q : \mathbb{R}^k \rightarrow \mathbb{R}$ uma função. Definimos o operador de diferença de q no intervalo I como sendo,

$$\Delta_I^k q(x_1, \dots, x_k) = q(x_1, \dots, x_{k-1}, b) - q(x_1, \dots, x_{k-1}, a) \quad (2.17)$$

Propriedade 2.21 A função de distribuição F de um vetor aleatório (X_1, \dots, X_n) satisfaz:

(i) $F(x_1, \dots, x_n)$ é não-decrescente em cada uma das suas variáveis. Logo, para cada coordenada $i = 1, \dots, n$, se $x < y$, então

$$F(x_1, \dots, x, \dots, x_n) \leq F(x_1, \dots, y, \dots, x_n); \quad (2.18)$$

(ii) $F(x_1, \dots, x_n)$ é contínua à direita em cada uma das suas variáveis. Logo, para cada coordenada $i = 1, \dots, n$, se $y_k \downarrow x_i$ quando $k \rightarrow \infty$, então

$$F(x_1, \dots, y_k, \dots, x_n) \downarrow F(x_1, \dots, x_i, \dots, x_n), \quad \text{quando } k \rightarrow \infty; \quad (2.19)$$

(iii) Para cada $1 \leq i \leq n$,

$$\lim_{x_i \rightarrow -\infty} F(x_1, \dots, x_n) = 0 \quad e \quad \lim_{x_i \rightarrow \infty} F(x_1, \dots, x_n) = 1; \quad (2.20)$$

(iv) Sejam cada $k = 1, \dots, n$,

$$\Delta_{I_1}^1 \dots \Delta_{I_n}^n F(x_1, \dots, x_n) \geq 0, \quad \forall I_k = (a_k, b_k], \quad (2.21)$$

onde $a_k < b_k$ e $\Delta_{I_1}^1 \dots \Delta_{I_n}^n$ é a composta dos operadores de diferença definidos em (2.20).

Observação 2.22 Para ilustrar a Propriedade 2.21-(iv), vejamos o caso em que $n = 2$. Seja F a função de distribuição do vetor aleatório (X, Y) . Se $I_1 = (a_1, b_1)$ e $I_2 = (a_2, b_2)$, então

$$\begin{aligned} \Delta_{I_1}^1 \Delta_{I_2}^2 F(x, y) &= \Delta_{I_1}^1 [F(x, b_2) - F(x, a_2)] = \\ &= F(b_1, b_2) - F(b_1, a_2) - [F(a_1, b_2) - F(a_1, a_2)] \geq 0. \end{aligned}$$

Definição 2.23 Uma função $F : \mathbb{R}^n \rightarrow \mathbb{R}$ que satisfaz as Propriedades 2.21(i) – (iv) é chamada função de distribuição n -dimensional (ou n -variada, ou multivariada).

Temos a seguinte analogia para a Definição 2.11-(b), no caso contínuo e multivariado.

Definição 2.24 Seja (X_1, \dots, X_n) um vetor aleatório e F a sua função de distribuição. Se existe uma função $g(x_1, \dots, x_n) \geq 0$ tal que

$$F(x_1, \dots, x_n) = \int_{-\infty}^{x_n} \dots \int_{-\infty}^{x_1} g(t_1, \dots, t_n) dt_1 \dots dt_n, \quad \forall (x_1, \dots, x_n) \in \mathbb{R}^n, \quad (2.22)$$

então g é a função de densidade do vetor aleatório (X_1, \dots, X_n) ou função de densidade conjunta das variáveis aleatórias X_1, \dots, X_n e, neste caso, dizemos que (X_1, \dots, X_n) é (absolutamente) contínuo.

Vejamos outras extensões de outros conceitos que são válidas para o caso multivariado.

Proposição 2.25 *Se $\tilde{X} = (X_1, \dots, X_n)$ é um vetor aleatório no espaço de probabilidade (Ω, \mathcal{F}, P) , então $[X \in \mathcal{B}] \in \mathcal{F}$, para todo $\mathcal{B} \in \mathcal{B}(\Omega)$, onde $\mathcal{B}(\Omega)$ é a σ -álgebra de Borel de $\Omega = \mathbb{R}^n$.*

Observação 2.26 *A σ -álgebra de Borel no \mathbb{R}^n é a menor σ -álgebra contendo todo retângulo n -dimensional, ou seja, é a σ -álgebra gerada pelos retângulos em \mathbb{R}^n .*

Definição 2.27 *Seja $\tilde{X} = (X_1, \dots, X_n)$ um vetor aleatório no espaço de probabilidade (Ω, \mathcal{F}, P) . A probabilidade definida em $\mathcal{B}(\Omega)$ por $P(\tilde{X} \in \mathcal{B})$ é chamada de função de distribuição de \tilde{X} ou função de distribuição conjunta de X_1, \dots, X_n .*

Proposição 2.28 *Se $\tilde{X} = (X_1, \dots, X_n)$ é um vetor aleatório contínuo, com densidade g , no espaço de probabilidade (Ω, \mathcal{F}, P) , então*

$$P(\tilde{X} \in \mathcal{B}) = \int \dots \int_{\mathcal{B}} g(x_1, \dots, x_n) dx_1, \dots, dx_n. \quad (2.23)$$

Proposição 2.29 [23, Prop. 2.5]

(a) *Se X_1, \dots, X_n são independentes e possuem densidades g_{X_1}, \dots, g_{X_n} , então a função*

$$g(x_1, \dots, x_n) = \prod_{i=1}^n g_{X_i}(x_i), \quad (2.24)$$

com $(x_1, \dots, x_n) \in \mathbb{R}^n$, é a densidade conjunta das variáveis aleatórias X_1, \dots, X_n , isto é, $g = g_{X_1, \dots, X_n}$.

(b) *Reciprocamente, se X_1, \dots, X_n têm densidade conjunta g satisfazendo*

$$g(x_1, \dots, x_n) = \prod_{i=1}^n g_i(x_i), \quad (2.25)$$

com $(x_1, \dots, x_n) \in \mathbb{R}^n$, onde $g_i(x) \geq 0$ e $\int_{-\infty}^{\infty} g_i(x) dx = 1$, para todo i , então X_1, \dots, X_n são independentes e $g_i = g_{X_i}$ (densidade de X_i), para $i = 1, \dots, n$.

Exemplo 2.30 (Distribuição uniforme [49]) *Sejam \mathcal{D} um conjunto compacto convexo com interior não vazio e μ a medida de Lebesgue. A distribuição uniforme nesse conjunto tem densidade definida por:*

$$g(x) = \begin{cases} \frac{1}{\mu(\mathcal{D})}, & x \in \mathcal{D} \\ 0, & x \notin \mathcal{D}. \end{cases} \quad (2.26)$$

2.1.4 Sequências de eventos e o Lema de Borel-Cantelli

Nesta seção vamos tratar do Lema de Borel-Cantelli, uma ferramenta das mais úteis na Teoria da Probabilidade, o qual sob determinada hipótese garante a ocorrência ou não de um evento infinitas vezes.

Se A_1, A_2, \dots é uma sequência de eventos, então o limite superior da sequência definido por

$$\limsup_{k \rightarrow \infty} A_k = \bigcap_{k=1}^{\infty} \bigcup_{j=k}^{\infty} A_j.$$

O evento $\limsup_{k \rightarrow \infty} A_k$ é o evento “ocorrência de um número infinito dos eventos A_k ”.

Usaremos a notação $\limsup_{k \rightarrow \infty} A_k = [A_k \text{ infinitas vezes}]$.

Proposição 2.31 (Lema de Borel-Cantelli) [23, Prop. 5.2]

Sejam A_1, A_2, \dots eventos aleatórios em (Ω, \mathcal{F}, P) , isto é, $A_k \in \mathcal{F}$ para todo k .

(i) *Se $\sum_{k=1}^{\infty} P(A_k) < \infty$, então $P(A_k \text{ infinitas vezes}) = 0$;*

(ii) *Se $\sum_{k=1}^{\infty} P(A_k) = \infty$ e os A_k 's são independentes, então $P(A_k \text{ infinitas vezes}) = 1$.*

2.1.5 Distribuição normal multivariada

Nesta seção, baseada em [23], estendemos a definição de distribuição normal para n dimensões.

Se $\tilde{X} = (X_1, \dots, X_n)$ é um vetor aleatório, onde X_1, \dots, X_n são variáveis aleatórias independentes com distribuição $N(0, 1)$, então o vetor aleatório $\tilde{Y} = (Y_1, \dots, Y_n)$, onde

$$Y_j = \gamma_j + a_{1j}X_1 + \dots + a_{nj}X_n,$$

para $j = 1, \dots, n$ possui distribuição normal n -variada (ou simplesmente, distribuição normal multivariada). As constantes a_{ij} e γ_j são números reais quaisquer. Desta forma, temos que \tilde{Y} é dado por

$$\tilde{Y} = \tilde{\gamma} + A\tilde{X}, \quad (2.27)$$

onde $\tilde{\gamma} = (\gamma_1, \dots, \gamma_n)$ e

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}. \quad (2.28)$$

Uma vez que Y_j é uma combinação linear de vetores aleatórios independentes com distribuição normal, segue que Y_j também possui distribuição normal, a saber

$$Y_j \sim N\left(\gamma_j, \sum_{k=1}^n a_{kj}^2\right).$$

A matriz de covariância de um vetor aleatório \tilde{Y} , a qual é simétrica e semidefinida positiva, é a matriz cujo elemento (i, j) é a covariância:

$$V = \text{Cov}(Y_i, Y_j) = (A^T A)_{ij}, \quad 1 \leq i \leq n \quad \text{e} \quad 1 \leq j \leq n, \quad (2.29)$$

onde A^T denota a matriz transposta de A .

Definição 2.32 Dizemos que os vetores aleatórios n -dimensionais $\tilde{X}_1, \tilde{X}_2, \dots$, definidos no mesmo espaço de probabilidade (Ω, \mathcal{F}, P) , são independentes e identicamente distribuídos se, e somente se,

$$P\left(\tilde{X}_1 \in \mathcal{B}_1, \dots, \tilde{X}_n \in \mathcal{B}_n\right) = \prod_{j=1}^n P(\tilde{X}_j \in \mathcal{B}_j), \quad (2.30)$$

para todo $n = 2, 3, \dots$ e para toda escolha dos Borelianos n -dimensionais $\mathcal{B}_1, \dots, \mathcal{B}_n$.

Evidentemente, tais vetores possuem mesma distribuição.

Definição 2.33 Sejam X_1, X_2, \dots, X_n variáveis aleatórias independentes e identicamente distribuídas tais que $X_j \sim N(0, 1)$ e \tilde{Y} o vetor aleatório obtido a partir de $\tilde{X} = (X_1, X_2, \dots, X_n)$ através da transformação

$$\tilde{Y} = \tilde{\gamma} + A\tilde{X}, \quad (2.31)$$

onde A é uma matriz real $n \times n$ e $\tilde{\gamma} \in \mathbb{R}^n$. Então dizemos que \tilde{Y} tem distribuição normal n -variada com média $\tilde{\gamma}$ e matriz de covariância $V = A^T A$, e denotamos por $\tilde{Y} \sim N(\tilde{\gamma}, V)$.

Além disso, se o posto (característica) da matriz A é igual a n , então \tilde{Y} possui densidade dada por

$$g(\tilde{y}) = (2\pi)^{-n/2} [\det(V)]^{-1/2} \Psi((\tilde{y} - \tilde{\gamma})^T V^{-1}(\tilde{y} - \tilde{\gamma})), \quad \tilde{y} \in \mathbb{R}^n, \quad (2.32)$$

onde V é a matriz de covariância $n \times n$, $\det(V)$ denota o determinante da matriz V e $\Psi(x) = e^{-x/2}$.

2.2 Convergência no contexto probabilístico

Um dos grandes interesses da teoria de probabilidade clássica e suas aplicações à Estatística está nos teoremas de limite, isto é, no comportamento assintótico de uma sequência de variáveis aleatórias. Tal sequência pode consistir de médias amostrais, somas cumulativas, extremos, quantis ou correlações amostrais e assim por diante. Existem várias noções para convergência das quais discutiremos aqui apenas as necessárias ao nosso trabalho, a *convergência quase certamente* e a *convergência em probabilidade*, bem como a relação entre esses dois tipos de convergência. Esta seção está baseada em [43, Cap. 6].

Definição 2.34 *Seja (Ω, \mathcal{F}, P) um espaço de probabilidade. Dizemos que uma declaração sobre elementos aleatórios ocorre quase certamente (q.s.) se a declaração ocorre a menos de um evento $\mathcal{N} \in \mathcal{F}$ tal que $P(\mathcal{N}) = 0$. Em outras palavras, tal declaração ocorre q.s. se existe um evento $\mathcal{N} \in \mathcal{F}$ com $P(\mathcal{N}) = 0$ e tal que a declaração ocorre sempre para $\omega \in \mathcal{N}^c$.*

Essas “declarações” podem ser igualdade ou desigualdade de variáveis aleatórias, limites de uma sequência de variáveis aleatórias, entre outros, como segue:

Exemplo 2.35 *Sejam X e Y duas variáveis aleatórias.*

- (a) $X = Y$ q.s. se existe um evento $\mathcal{N} \in \mathcal{F}$ com $P(\mathcal{N}) = 0$ e tal que $X(\omega) = Y(\omega)$ para todo $\omega \in \mathcal{N}^c$.

(b) $X \leq Y$ q.s. se existe um evento $\mathcal{N} \in \mathcal{F}$ com $P(\mathcal{N}) = 0$ e tal que $X(\omega) \leq Y(\omega)$ para todo $\omega \in \mathcal{N}^c$.

Exemplo 2.36 Se (X_k) é uma sequência de variáveis aleatórias, então $\lim_{k \rightarrow \infty} X_k$ existe q.s. se existe um evento $\mathcal{N} \in \mathcal{F}$ com $P(\mathcal{N}) = 0$ e tal que $\lim_{k \rightarrow \infty} X_k(\omega)$ existe para todo $\omega \in \mathcal{N}^c$.

Definição 2.37 Sejam X_k , com $k \geq 1$, e X^* variáveis aleatórias. Dizemos que a sequência (X_k) converge em probabilidade para X^* , e denotamos $X_k \xrightarrow{P} X^*$, se para todo $\varepsilon > 0$ tem-se que $\lim_{k \rightarrow \infty} P(\|X_k - X^*\| > \varepsilon) = 0$, onde $\|\cdot\|$ é uma norma arbitrária.

Convergência em probabilidade é um conceito mais fraco de convergência, no sentido de que é exigido uma proximidade apenas em termos da probabilidade. O teorema a seguir estabelece que convergir q.s. implica em convergir em probabilidade.

Teorema 2.38 [43, Teo. 6.2.1] Sejam (X_k) e X^* variáveis aleatórias num espaço de probabilidade (Ω, \mathcal{F}, P) . Se $X_k \rightarrow X^*$ q.s., então $X_k \xrightarrow{P} X^*$.

Porém a recíproca do Teorema 2.38 não é verdadeira [43, Exemplo 6.2.1].

Definição 2.39 Uma sequência (X_k) de elementos aleatórios reais é dita ser Cauchy em probabilidade se para qualquer $\varepsilon > 0$, existe $k_0 = k_0(\varepsilon, \delta)$ tal que para todo $t, s \geq k_0$ temos que $P(\|X_t - X_s\| > \varepsilon) < \delta$. Ou seja, $(X_k)_{k \geq 1}$ é Cauchy em probabilidade se $(X_k - X_q) \xrightarrow{P} 0$ quando $k, q \rightarrow \infty$.

Teorema 2.40 [43, Teor. 6.3.1] Sejam (X_k) e X^* variáveis aleatórias de valores reais.

- (i) Critério de Cauchy: (X_k) converge em probabilidade se, e somente se, (X_k) é Cauchy em probabilidade.
- (ii) $X_k \xrightarrow{P} X^*$ se, e somente se, cada subsequência X_{k_j} contém uma outra subsequência $X_{k_{j(\ell)}}$ que converge q.s. para X^* .

Capítulo 3

Algoritmos estocásticos com iterandos viáveis

O objetivo neste capítulo é estudar a convergência, no sentido probabilístico, de métodos estocásticos com iterandos viáveis para otimização global de uma função determinística, sujeita apenas a restrições do tipo limites simples nas suas variáveis de decisão. Um algoritmo estocástico de busca direta bastante geral e com iterandos viáveis, baseado nas ideias de [38], é apresentado. Em seguida, é efetuado um estudo das condições suficientes para que tal algoritmo seja convergente. Este capítulo serve de base para o estudo futuro acerca da convergência de um algoritmo estocástico de busca direta geral para resolver problemas de otimização não lineares com restrições gerais, o qual trabalha com iterandos inviáveis e em que as restrições do problema serão tratadas pelo método de filtro.

3.1 O problema

Considere-se o problema de otimização escrito da seguinte forma

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a} & x \in \mathcal{D}, \end{array} \tag{3.1}$$

onde $f : A \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função determinística qualquer e $\mathcal{D} \subseteq A \subseteq \mathbb{R}^n$ um conjunto não vazio.

Suponha que a função f é limitada inferiormente em \mathcal{D} , ou seja, $f^* := \inf_{x \in \mathcal{D}} f(x) > -\infty$. Dizemos que x^* é um minimizador global de f sobre o conjunto \mathcal{D} quando $f(x^*) \leq f(x)$ para todo $x \in \mathcal{D}$. Se x^* existe temos $f^* = f(x^*)$. Um minimizador global de f sobre \mathcal{D} pode existir mesmo que \mathcal{D} não seja um conjunto compacto ou que f seja descontínua em certas regiões de \mathcal{D} . Por outro lado, o problema de otimização global (3.1) pode ter múltiplos minimizadores globais.

3.2 Um algoritmo geral

Nesta seção vamos apresentar um algoritmo geral estocástico com iterandos viáveis para resolver o problema (3.1) e estudar sua convergência no sentido probabilístico.

3.2.1 Algoritmo

Devido ao perfil estocástico do algoritmo em estudo, os iterandos são tratados como vetores aleatórios n -dimensionais cujas realizações estão em $\mathcal{D} \subseteq \mathbb{R}^n$. Em outras palavras, os iterandos gerados pelo algoritmo são dados por uma sequência de vetores aleatórios $(X_k)_{k \geq 1}$ definidos num espaço de probabilidade (Ω, \mathcal{F}, P) , onde Ω é o conjunto de resultados, \mathcal{F} é uma σ -álgebra e $P : \mathcal{F} \rightarrow [0, 1]$ é uma função que determina a probabilidade dos eventos.

Em cada iteração k , para determinar o vetor aleatório $X_k \in \mathcal{D}$ será utilizada uma coleção de elementos aleatórios intermediários, denotada por $\{\Lambda_{k_j} : (\Omega, \mathcal{F}) \rightarrow (\Omega_{k_j}, \mathcal{F}_{k_j}), \text{ com } k \geq 1 \text{ e } j = 1, 2, \dots, \ell_k\}$. Tais Λ_{k_j} podem ser variáveis aleatórias ou vetores aleatórios ou quaisquer outros tipos de elementos aleatórios definidos no espaço de probabilidade (Ω, \mathcal{F}, P) .

Dado um vetor aleatório inicial $X_0 \in \mathcal{D}$, considere um *oráculo* associado à coleção de elementos aleatórios gerada até a iteração k , o qual é definido da seguinte maneira:

$\mathcal{O}_0 = \{X_0\}$ e para cada $k \geq 1$,

$$\mathcal{O}_k := \{\Lambda_{t_j} : t = 1, 2, \dots, k \text{ e } j = 1, 2, \dots, \ell_k\} = \mathcal{O}_{k-1} \cup \{\Lambda_{k1}, \Lambda_{k2}, \dots, \Lambda_{k\ell_k}\} \quad (3.2)$$

Para $k \geq 0$, defina $\sigma(\mathcal{O}_k)$ como sendo uma σ -álgebra gerada pelos elementos aleatórios do oráculo \mathcal{O}_k . Além disso, $\sigma(\mathcal{O}_k)$ pode ser vista como uma representação de todas as informações que podem ser calculadas a partir dos elementos aleatórios de \mathcal{O}_k .

Fixada uma iteração k , denotaremos por $\Theta : \mathcal{O}_k \rightarrow \mathcal{D}$ uma função determinística dos elementos aleatórios de \mathcal{O}_k e definimos $X_k = \Theta(\mathcal{O}_k)$. Isto é, X_k é uma função determinística dos elementos aleatórios de \mathcal{O}_k .

A seguir apresentamos o algoritmo estocástico geral com iterandos viáveis, o qual é uma adaptação do algoritmo geral dado em [38].

Algoritmo 3.1 - Algoritmo Geral

DADOS: $X_0 \in \mathcal{D}$, $X_0^* = X_0$ e $\mathcal{O}_0 = \{X_0^*\}$

$k = 1$

ENQUANTO O CRITÉRIO DE PARADA NÃO FOR SATISFEITO

Gerar uma realização para cada elemento intermediário aleatório de acordo com alguma distribuição de probabilidade: $\{\Lambda_{kj} : 1 \leq j \leq \ell_k\}$

Defina $\mathcal{O}_k = \mathcal{O}_{k-1} \cup \{\Lambda_{kj} : 1 \leq j \leq \ell_k\}$

$X_k = \Theta(\mathcal{O}_k)$, onde assume-se que qualquer realização sua é um ponto viável

SE $f(X_k) < f(X_{k-1}^*)$

$$X_k^* = X_k$$

SENÃO

$$X_k^* = X_{k-1}^*$$

$k = k + 1$

Note que no Algoritmo Geral são geradas uma sequência (X_k) de iterandos (i.e. pontos de avaliação da função) obtidos aleatoriamente e uma sequência (X_k^*) dos melhores pontos, tal que a sequência real $(f(X_k^*))$ é monótona não crescente.

Dizemos que o Algoritmo Geral converge para o mínimo global de f em \mathcal{D} em probabilidade (ou quase certamente) se a sequência $(f(X_k^*))$ gerada pelo algoritmo converge para f^* em probabilidade (ou quase certamente).

3.2.2 Condições suficientes para convergência do algoritmo

Agora vamos discutir condições que garantem a convergência do Algoritmo Geral no sentido probabilístico, já que se trata de um algoritmo estocástico. O teorema a seguir, que é uma releitura de [38, Teoremas 1 e 2], apresenta uma condição suficiente para tal convergência.

Teorema 3.1 *Suponhamos que o Algoritmo Geral satisfaz a seguinte propriedade:*

Para todo $\varepsilon > 0$, existe $L(\varepsilon) \in (0, 1)$ tal que

$$P(X_{k_j} \in \mathcal{D} : f(X_{k_j}) < f^* + \varepsilon \mid \sigma(\mathcal{O}_{(k_j)-1})) \geq L(\varepsilon), \quad (3.3)$$

para alguma subsequência de índices $\{k_j\}_{j \geq 1}$. Então $f(X_k^) \rightarrow f^*$ quase certamente. Além disso, se o minimizador x^* existe e é único então $X_k^* \rightarrow x^*$ quase certamente.*

Demonstração. Fixado $\varepsilon > 0$, defina $S_\varepsilon := \{x \in \mathcal{D} : f(x) < f^* + \varepsilon\}$. Pela Hipótese (3.3) tem-se, para qualquer $j \geq 1$

$$P(X_{k_j} \in S_\varepsilon \mid \sigma(\mathcal{O}_{(k_j)-1})) \geq L(\varepsilon). \quad (3.4)$$

Agora, para cada $j \geq 1$, de (2.4) e da Propriedade 2.4 (a), segue

$$\begin{aligned} P(X_{k_1} \notin S_\varepsilon, \dots, X_{k_j} \notin S_\varepsilon) &= \prod_{i=1}^j P(X_{k_i} \notin S_\varepsilon \mid X_{k_1} \notin S_\varepsilon, \dots, X_{k_{(i-1)}} \notin S_\varepsilon) \\ &= \prod_{i=1}^j \left(1 - P(X_{k_i} \in S_\varepsilon \mid X_{k_1} \notin S_\varepsilon, \dots, X_{k_{(i-1)}} \notin S_\varepsilon)\right). \end{aligned} \quad (3.5)$$

Por outro lado, de (3.4) e do condicionamento dos elementos aleatórios em $\mathcal{O}_{(k_i)-1}$ tem-se que

$$P(X_{k_i} \in S_\varepsilon \mid X_{k_1} \notin S_\varepsilon, X_{k_2} \notin S_\varepsilon, \dots, X_{k_{(i-1)}} \notin S_\varepsilon) \geq L(\varepsilon). \quad (3.6)$$

Portanto, de (3.5) e (3.6) segue que

$$P(X_{k_1} \notin S_\varepsilon, X_{k_2} \notin S_\varepsilon, \dots, X_{k_j} \notin S_\varepsilon) \leq (1 - L(\varepsilon))^j. \quad (3.7)$$

Observe que se \bar{k} é o menor índice tal que $X_{\bar{k}} \in S_\varepsilon$, atendendo à forma como a sequência (X_k^*) é definida no Algoritmo Geral, então tem-se que $X_{\bar{k}}^* = X_{\bar{k}}$ e que, para todo $k \geq \bar{k}$,

$$f(X_k^*) \leq f(X_{k-1}^*) \leq \dots \leq f(X_{\bar{k}}^*) < f^* + \varepsilon.$$

Ou seja, $X_k^* \in S_\varepsilon$ para todo $k \geq \bar{k}$. Consequentemente,

$$X_{k_j}^* \notin S_\varepsilon \Rightarrow X_{k_1} \notin S_\varepsilon, X_{k_2} \notin S_\varepsilon, \dots, X_{k_j} \notin S_\varepsilon. \quad (3.8)$$

Logo, de (3.7), (3.8) e da Propriedade 2.4-(c), para cada $j \geq 1$, tem-se

$$\begin{aligned} 0 &\leq P(f(X_{k_j}^*) - f^* \geq \varepsilon) = P(f(X_{k_j}^*) \geq f^* + \varepsilon) = P(X_{k_j}^* \notin S_\varepsilon) \leq \\ &\leq P(X_{k_1} \notin S_\varepsilon, X_{k_2} \notin S_\varepsilon, \dots, X_{k_j} \notin S_\varepsilon) \leq (1 - L(\varepsilon))^j. \end{aligned}$$

Portanto,

$$\lim_{j \rightarrow \infty} P(f(X_{k_j}^*) - f^* \geq \varepsilon) = 0.$$

Ou seja, $f(X_{k_j}^*) \rightarrow f^*$ em probabilidade. E pelo item (b) do Teorema 2.40 segue que $f(X_{k_{j(i)}}^*) \rightarrow f^*$ quase certamente, com $i \rightarrow \infty$ para alguma subsequência de índices $\{k_{j(i)}\}_{i \geq 1}$.

Dado que $f^* > -\infty$ e a sequência $(f(X_k^*))_{k \geq 1}$ é monótona não crescente, tem-se que $\lim_{k \rightarrow \infty} f(X_k^*(\omega))$ existe para todo ponto amostral. Assim, da unicidade do limite podemos concluir que

$$f(X_k^*) \rightarrow f^* \quad \text{q.s.} \quad (3.9)$$

Resta agora mostrar que, se x^* existe e é único então $X_k^* \rightarrow x^*$ quase certamente. Para isso, fixado $\varepsilon > 0$ defina $\tilde{f} := \inf_{x \in \mathcal{D}, \|x - x^*\| \geq \varepsilon} f(x)$. Da unicidade de x^* , segue que $\tilde{f} > f^*$. De (3.9) temos que tal convergência ocorre a menos de um conjunto de medida nula; isto é, existe $\mathcal{N} \subset \Omega$ com $P(\mathcal{N}) = 0$ e tal que $f(X_k^*(\omega)) \rightarrow f(x^*)$, para todo $\omega \in \mathcal{N}^c$. Logo, da

definição de limite de sequência, para todo $\omega \in \mathcal{N}^c$ existe um inteiro N tal que para todo $k \geq N$,

$$f(X_k^*(\omega)) - f(x^*) = |f(X_k^*(\omega)) - f(x^*)| < \tilde{f} - f(x^*),$$

ou seja,

$$f(X_k^*(\omega)) < \tilde{f}, \quad (3.10)$$

para todo $k \geq N$ e $\omega \in \mathcal{N}^c$.

Agora vamos mostrar que $\|X_k^*(\omega) - x^*\| < \varepsilon$ para todo $\omega \in \mathcal{N}^c$. De fato, suponha por contradição que existe $\omega \in \mathcal{N}^c$ tal que $\|X_k^*(\omega) - x^*\| \geq \varepsilon$. Da definição de \tilde{f} , segue que $f(X_k^*(\omega)) \geq \tilde{f}$, o que contradiz (3.10). Assim, temos que para todo $\omega \in \mathcal{N}^c$ existe um inteiro N tal que para todo $k \geq N$, $\|X_k^*(\omega) - x^*\| < \varepsilon$. Portanto, $X_k^*(\omega) \rightarrow x^*$ q.s. \square

O lema a seguir nos auxiliará na demonstração dos próximos teoremas.

Lema 3.2 *Sejam μ a medida de Lebesgue em \mathbb{R}^n e x^* um minimizador global de f sobre \mathcal{D} , com f contínua em x^* . Considere que para todo $\delta > 0$,*

$$\psi_{\mathcal{D}}(\delta) := \inf_{z \in \mathcal{D}} \mu(B(z, \delta) \cap \mathcal{D}) > 0,$$

e seja $(X_{k_j})_{j \geq 1}$ uma subsequência de (X_k) tal que, para cada $j \geq 1$, X_{k_j} tem uma densidade condicional denotada por $g_{k_j}(x | \sigma(\mathcal{O}_{(k_j)-1}))$.

(i) *Se*

$$\mu(\{x \in \mathcal{D} : G(x) = 0\}) = 0, \quad \text{onde } G(x) := \inf_{j \geq 1} g_{k_j}(x | \sigma(\mathcal{O}_{(k_j)-1})), \quad (3.11)$$

então para qualquer $z \in \mathcal{D}$ e $\delta > 0$, existe $v(z, \delta) \in (0, 1)$ tal que

$$P(X_{k_j} \in B(z, \delta) \cap \mathcal{D} | \sigma(\mathcal{O}_{(k_j)-1})) \geq v(z, \delta). \quad (3.12)$$

(ii) *Nas mesmas condições de (i), para todo $\varepsilon > 0$, existe $L(\varepsilon) \in (0, 1)$ tal que*

$$P(X_{k_j} \in \mathcal{D} : f(X_{k_j}) < f(x^*) + \varepsilon | \sigma(\mathcal{O}_{(k_j)-1})) \geq L(\varepsilon).$$

Demonstração.

- (i) Fixados $\delta > 0$ e $z \in \mathcal{D}$, da definição de probabilidade e das propriedades de integração temos, para todo $j \geq 1$, que

$$\begin{aligned} P(X_{k_j} \in (B(z, \delta) \cap \mathcal{D}) \mid \sigma(\mathcal{O}_{(k_j)-1})) &= \int_{B(z, \delta) \cap \mathcal{D}} g_{k_j}(x \mid \sigma(\mathcal{O}_{(k_j)-1})) dx \\ &\geq \int_{B(z, \delta) \cap \mathcal{D}} G(x) dx. \end{aligned}$$

Uma vez que G é não negativa em \mathcal{D} , $\mu(\{x \in \mathcal{D} : G(x) = 0\}) = 0$, e $\mu(B(z, \delta) \cap \mathcal{D}) \geq \psi_{\mathcal{D}}(\delta) > 0$, segue que

$$v(z, \delta) := \int_{B(z, \delta) \cap \mathcal{D}} G(x) dx > 0.$$

Logo, para todo $j \geq 1$, existe $v(z, \delta) \in (0, 1)$, tal que

$$P(X_{k_j} \in (B(z, \delta) \cap \mathcal{D}) \mid \sigma(\mathcal{O}_{(k_j)-1})) \geq v(z, \delta).$$

- (ii) Fixado $\varepsilon > 0$, da continuidade de f em x^* existe $\delta(\varepsilon) > 0$ tal que se $\|x - x^*\| < \delta(\varepsilon)$, então $|f(x) - f(x^*)| < \varepsilon$. Consequentemente, temos a seguinte relação entre os eventos, para cada $j \geq 1$,

$$\begin{aligned} [X_{k_j} \in \mathcal{D} : f(X_{k_j}) < f(x^*) + \varepsilon] &= [X_{k_j} \in \mathcal{D} : |f(X_{k_j}) - f(x^*)| < \varepsilon] \\ &\supseteq [X_{k_j} \in \mathcal{D} : \|X_{k_j} - x^*\| < \delta(\varepsilon)]. \end{aligned}$$

Assim, da propriedade de monotonicidade (2.4) e da hipótese (3.12),

$$\begin{aligned} P(X_{k_j} \in \mathcal{D} : f(X_{k_j}) < f(x^*) + \varepsilon \mid \sigma(\mathcal{O}_{(k_j)-1})) &\geq \\ &\geq P(X_{k_j} \in \mathcal{D} : \|X_{k_j} - x^*\| < \delta(\varepsilon) \mid \sigma(\mathcal{O}_{(k_j)-1})) \geq v(x^*, \delta(\varepsilon)) =: L(\varepsilon) > 0, \end{aligned}$$

□

O teorema a seguir, uma releitura de [38, Teorema 3], estuda uma condição suficiente para a convergência no caso em que f é contínua num minimizador global de f sobre \mathcal{D} .

Teorema 3.3 *Seja x^* um minimizador global de f sobre \mathcal{D} e suponha que f é contínua em x^* . Mais ainda, suponha que o Algoritmo Geral satisfaz a seguinte propriedade:*

Para qualquer $z \in \mathcal{D}$ e $\delta > 0$, existe $v(z, \delta) \in (0, 1)$ tal que

$$P(X_{k_j} \in B(z, \delta) \cap \mathcal{D} \mid \sigma(\mathcal{O}_{(k_j)-1})) \geq v(z, \delta), \quad (3.13)$$

para alguma subsequência de índices $\{k_j\}_{j \geq 1}$, onde $B(z, \delta)$ é uma bola aberta centrada em z e com raio δ . Então $f(X_k^*) \rightarrow f^*$ quase certamente. Além disso, se x^* é o único minimizador global, então $X_k^* \rightarrow x^*$ quase certamente.

Demonstração. Pelo item (ii) do Lema 3.2 temos que para todo $\varepsilon > 0$, existe $L(\varepsilon) \in (0, 1)$ tal que,

$$P(X_{k_j} \in \mathcal{D} : f(X_{k_j}) < f(x^*) + \varepsilon \mid \sigma(\mathcal{O}_{(k_j)-1})) \geq L(\varepsilon).$$

Portanto, o resultado segue do Teorema 3.1. \square

Os dois teoremas a seguir, os quais são uma adaptação de [38, Teoremas 4 e 5] para o Algoritmo Geral, apresentam uma condição suficiente para a garantia da convergência do algoritmo em termos do ínfimo das funções densidade condicional dos iterandos de vetores aleatórios candidatos/tentativos.

Teorema 3.4 *Sejam μ a medida de Lebesgue em \mathbb{R}^n , x^* um minimizador global de f sobre \mathcal{D} , com f contínua em x^* , e considere as sequências (X_k) e (X_k^*) geradas pelo Algoritmo Geral. Considere ainda que para todo $\delta > 0$,*

$$\psi_{\mathcal{D}}(\delta) := \inf_{z \in \mathcal{D}} \mu(B(z, \delta) \cap \mathcal{D}) > 0.$$

Suponha que existe uma subsequência de índices $\{k_j\}_{j \geq 1}$ tal que, para cada $j \geq 1$, X_{k_j} tem uma densidade condicional $g_{k_j}(x \mid \sigma(\mathcal{O}_{(k_j)-1}))$ satisfazendo a seguinte condição:

$$\mu(\{x \in \mathcal{D} : G(x) = 0\}) = 0, \quad \text{onde} \quad G(x) := \inf_{j \geq 1} g_{k_j}(x \mid \sigma(\mathcal{O}_{(k_j)-1})). \quad (3.14)$$

Então, $f(X_k^) \rightarrow f^*$ quase certamente. Além disso, se x^* é o único minimizador global, então $X_k^* \rightarrow x^*$ quase certamente.*

Demonstração. Pelo item (i) do Lema 3.2 temos que para todo $\delta > 0$ e $z \in \mathcal{D}$, existe $v(z, \delta) \in (0, 1)$ tal que,

$$P(X_{k_j} \in (B(z, \delta) \cap \mathcal{D}) \mid \sigma(\mathcal{O}_{(k_j)-1})) \geq v(z, \delta).$$

Portanto, do teorema anterior seguem os resultados. \square

O teorema a seguir contempla a situação em que os iterandos de vetores aleatórios são gerados usando diferentes distribuições de probabilidade nas diferentes componentes de um iterando. Dado um vetor aleatório X , vamos denotar por $(X)_i$ a variável aleatória que corresponde à i -ésima coordenada de X e por $(v)_i$ a i -ésima coordenada de um ponto $v \in D$. Seja $(D)_i$ o conjunto definido por $(D)_i := \{x \in \mathbb{R} : x = (v)_i \text{ para algum } v \in D\}$.

Teorema 3.5 *Sejam μ a medida de Lebesgue em \mathbb{R}^n , x^* um minimizador global de f sobre \mathcal{D} , com f contínua em x^* . Considere as sequências (X_k) e (X_k^*) geradas pelo Algoritmo Geral. Suponha que $\psi_{\mathcal{D}}(\delta) := \inf_{z \in \mathcal{D}} \mu(B(z, \delta) \cap \mathcal{D}) > 0$ para todo $\delta > 0$, e que existe uma subsequência de índices $\{k_j\}_{j \geq 1}$ tal que as seguintes propriedades são satisfeitas:*

- (i) *Para cada $j \geq 1$ as variáveis aleatórias $(X_{k_j})_1, (X_{k_j})_2, \dots, (X_{k_j})_n$ são condicionalmente independentes dados os elementos aleatórios em $\mathcal{O}_{(k_j)-1}$; e*
- (ii) *Para cada $j \geq 1$ e para cada $1 \leq i \leq n$, a variável aleatória $(X_{k_j})_i$ tem uma densidade condicional $(g_{k_j})_i(u \mid \sigma(\mathcal{O}_{(k_j)-1}))$ que satisfaz:*

$$\mu(\{u \in (\mathcal{D})_i : G_i(u) = 0\}) = 0, \quad \text{onde} \quad G_i(u) := \inf_{j \geq 1} (g_{k_j})_i(u \mid \sigma(\mathcal{O}_{(k_j)-1})). \quad (3.15)$$

Então, $f(X_k^*) \rightarrow f^*$ quase certamente. Além disso, se x^* é o único minimizador global, então $X_k^* \rightarrow x^*$ quase certamente.

Demonstração. Da propriedade (i) temos que, para cada $j \geq 1$, X_{k_j} tem uma densidade condicional dada por

$$g_{k_j}(x \mid \sigma(\mathcal{O}_{(k_j)-1})) = \prod_{i=1}^n (g_{k_j})_i((x)_i \mid \sigma(\mathcal{O}_{(k_j)-1})).$$

Além disso, da definição de G e de propriedade de ínfimo segue que

$$G(x) = \inf_{j \geq 1} \prod_{i=1}^n (g_{k_j})_i((x)_i \mid \sigma(\mathcal{O}_{(k_j)-1})) \geq \prod_{i=1}^n \left(\inf_{j \geq 1} (g_{k_j})_i((x)_i \mid \sigma(\mathcal{O}_{(k_j)-1})) \right) = \prod_{i=1}^n G_i((x)_i).$$

Uma vez que

$$\{x \in \mathcal{D} : G(x) = 0\} \subseteq \bigcup_{i=1}^n \{x \in \mathcal{D} : G_i((x)_i) = 0\}$$

temos

$$\begin{aligned} \mu(\{x \in \mathcal{D} : G(x) = 0\}) &\leq \mu\left(\bigcup_{i=1}^n \{x \in \mathcal{D} : G_i((x)_i) = 0\}\right) \\ &\leq \sum_{i=1}^n \mu(\{x \in \mathcal{D} : G_i((x)_i) = 0\}). \end{aligned} \quad (3.16)$$

Além disso,

$$\{x \in \mathcal{D} : G_i((x)_i) = 0\} \subseteq (\mathcal{D})_1 \times \cdots \times \{v \in (\mathcal{D})_i : G_i(v) = 0\} \times \cdots \times (\mathcal{D})_n,$$

usando a propriedade (ii) segue que,

$$\begin{aligned} \mu(\{x \in \mathcal{D} : G_i((x)_i) = 0\}) &\leq \mu((\mathcal{D})_1 \times \cdots \times \{v \in (\mathcal{D})_i : G_i(v) = 0\} \times \cdots \times (\mathcal{D})_n) = \\ &= \mu((\mathcal{D})_1) \cdots \mu((\mathcal{D})_{i-1}) \mu(\{v \in (\mathcal{D})_i : G_i(v) = 0\}) \mu((\mathcal{D})_{i+1}) \cdots \mu((\mathcal{D})_n) = 0. \end{aligned}$$

Usando isto em (3.16), $\mu(\{x \in \mathcal{D} : G(x) = 0\}) = 0$ e pelo Teorema 3.4 seguem os resultados. \square

O próximo teorema, uma releitura de [38, Teor. 6 e Cor. 2], garante a convergência em probabilidade do Algoritmo Geral quando existe uma subsequência em que os iterandos são dados como a soma do melhor ponto obtido, até então, com um vetor aleatório cuja distribuição é multivariada normal.

Teorema 3.6 *Seja μ a medida de Lebesgue em \mathbb{R}^n . Suponha que \mathcal{D} é um conjunto limitado satisfazendo $\psi_{\mathcal{D}}(\delta) := \inf_{z \in \mathcal{D}} \mu(B(z, \delta) \cap \mathcal{D}) > 0$ para todo $\delta > 0$. Seja x^* um minimizador global de f sobre \mathcal{D} , com f contínua em x^* . Considere as sequências (X_k) e (X_k^*) geradas pelo Algoritmo Geral e suponha que existe uma subsequência de índices $\{k_j\}_{j \geq 1}$ tal que para cada $j \geq 1$, tem-se*

$$X_{k_j} = X_{(k_j)-1}^* + W_j, \quad (3.17)$$

onde W_j é um vetor aleatório cuja distribuição condicional dado $\sigma(\mathcal{O}_{(k_j)-1})$ é multivariada normal com vetor de média nulo e matriz de covariância V_j . Para cada $j \geq 1$, seja $\bar{\lambda}_j$ o menor autovalor de V_j . Se $\inf_{j \geq 1} \bar{\lambda}_j > 0$, então $f(X_k^*) \rightarrow f^*$ quase certamente. Além disso, se x^* é o único minimizador global de f em \mathcal{D} , então $X_k^* \rightarrow x^*$ quase certamente.

Demonstração. Pelo mecanismo do algoritmo,

$$X_{(k_j)-1}^* = \sum_{i=1}^{(k_j)-1} X_i 1_E(X_i), \quad (3.18)$$

onde 1_E é a função indicadora e E é o evento definido por

$$E = [f(X_\ell) \leq f(X_t), \forall t = 1, \dots, (k_j) - 1 \text{ e } \ell \text{ é o menor índice com esta propriedade}].$$

Para cada $i = 1, \dots, (k_j) - 1$, X_i é uma função determinística de elementos aleatórios em \mathcal{O}_i . Consequentemente, $X_{(k_j)-1}^*$ é uma função determinística dos elementos aleatórios em $\mathcal{O}_{(k_j)-1}$. Sendo assim, por (3.17) a distribuição condicional de X_{k_j} dado $\sigma(\mathcal{O}_{(k_j)-1})$ é uma distribuição multivariada normal, com média $X_{(k_j)-1}^*$ e matriz de covariância V_j . Esta matriz é invertível para todo $j \geq 1$, pois por hipótese $\inf_{j \geq 1} \bar{\lambda}_j > 0$. Logo, por (2.32), X_{k_j} tem densidade condicional dada por

$$g_{k_j}(x \mid \sigma(\mathcal{O}_{(k_j)-1})) = \gamma [\det(V_j)]^{-1/2} \Psi \left((x - X_{(k_j)-1}^*)^T V_j^{-1} (x - X_{(k_j)-1}^*) \right), \quad (3.19)$$

onde $\Psi(x) = e^{-x/2}$, $\gamma = (2\pi)^{-n/2}$ e $x \in R^n$.

Da definição da norma Euclidiana e de $\bar{\lambda}_j$ segue que,

$$\|V_j^{-1}\| = \sqrt{\text{maior autovalor de } (V_j^{-1})^T V_j^{-1}} = \sqrt{\frac{1}{\bar{\lambda}_j^2}} = \frac{1}{|\bar{\lambda}_j|} = \frac{1}{\bar{\lambda}_j}.$$

Usando o fato que V_j é definida positiva, a desigualdade de Cauchy-Schwarz e a igualdade anterior temos, para cada $j \geq 1$ e $x \in \mathcal{D}$,

$$(x - X_{(k_j)-1}^*)^T V_j^{-1} (x - X_{(k_j)-1}^*) \leq \|x - X_{(k_j)-1}^*\|^2 \|V_j^{-1}\| \leq \frac{D^2}{\bar{\lambda}_j},$$

onde $D = \text{diam}(\mathcal{D}) = \sup_{x, y \in \mathcal{D}} \|x - y\| < \infty$, uma vez que \mathcal{D} é limitado. Como Ψ é monotonamente não crescente,

$$\Psi \left((x - X_{(k_j)-1}^*)^T V_j^{-1} (x - X_{(k_j)-1}^*) \right) \geq \Psi \left(\frac{D^2}{\bar{\lambda}_j} \right). \quad (3.20)$$

Uma vez que o determinante de uma matriz é o produto dos seus autovalores, segue que $\det(V_j) \leq (\lambda_j^*)^n$, onde λ_j^* é o maior autovalor de V_j . Usando isto e (3.20) em (3.19) e o

fato de que Ψ é monotonicamente não crescente temos, para cada $x \in \mathcal{D}$

$$g_{k_j}(x \mid \sigma(\mathcal{O}_{(k_j)-1})) \geq \gamma (\lambda_j^*)^{-n/2} \Psi \left(\frac{D^2}{\bar{\lambda}_j} \right) \geq \gamma \left(\sup_{\ell \geq 1} \lambda_\ell^* \right)^{-n/2} \Psi \left(\frac{D^2}{\inf_{\ell \geq 1} \bar{\lambda}_\ell} \right).$$

Logo, para todo $x \in \mathcal{D}$

$$G(x) = \inf_{j \geq 1} g_{k_j}(x \mid \sigma(\mathcal{O}_{(k_j)-1})) \geq \gamma \left(\sup_{\ell \geq 1} \lambda_\ell^* \right)^{-n/2} \Psi \left(\frac{D^2}{\inf_{\ell \geq 1} \bar{\lambda}_\ell} \right) > 0,$$

o que implica,

$$\mu(\{x \in \mathcal{D} : G(x) = 0\}) = 0.$$

E, assim, do Teorema 3.4 seguem os resultados. \square

Portanto, qualquer algoritmo estocástico nos moldes do Algoritmo Geral satisfazendo alguma das condições suficientes estudadas nesta seção converge em probabilidade.

3.3 Exemplos de aplicação da teoria de convergência

Nesta seção apresentam-se três algoritmos estocásticos práticos, para resolver problemas do tipo (3.1), onde o conjunto viável \mathcal{D} tem a propriedade de que é fácil calcular a projeção de um ponto sobre \mathcal{D} . A projeção ortogonal de $x \in \mathbb{R}^n$ sobre \mathcal{D} será denotada por $proj_{\mathcal{D}}(x)$. Aplicaremos, também, o estudo da convergência feito na seção anterior.

3.3.1 Algoritmo de busca aleatória localizada

Algoritmos de busca aleatória localizada permitem uma exploração da informação previamente obtida sobre a função objetivo no seu domínio \mathcal{D} . Tais algoritmos são muitas vezes conhecidos como algoritmos localizados para enfatizar a sua dependência em relação à região local perto da estimativa corrente numa determinada iteração. Portanto, esta terminologia não deve ser confundida com busca de uma solução global ou local para o problema de otimização. Para uma leitura mais completa acerca do assunto, consultar [24, 31, 50].

Vamos estudar um caso especial do algoritmo geral de busca aleatória localizada que envolve a distribuição normal multivariada, bem como a sua convergência em probabilidade. Esta seção está baseada em [38, Sec. 5.1], onde se encontra a versão original do *Algoritmo de busca aleatória localizada* (do inglês, *Localized Random Search* - LRS) para o caso de distribuição normal multivariada. Tal algoritmo resolve problemas do tipo (3.1), onde \mathcal{D} é um conjunto limitado, e seus iterandos X_k são obtidos adicionando-se perturbações aleatórias, com uma distribuição normal multivariada, às componentes do melhor ponto corrente.

Com o objetivo de colocar o Algoritmo LRS nos moldes do Algoritmo Geral, dado na Seção 3.2, vamos estabelecer como deve ser gerado o conjunto dos elementos aleatórios $\{\Lambda_{k_j}\}_{1 \leq j \leq \ell_k}$ e a função determinística Θ .

Coleção dos elementos aleatórios

Dado $X_0 \in \mathcal{D}$, a coleção de elementos aleatórios $\{\Lambda_{k_j}\}_{1 \leq j \leq \ell_k}$ utilizada no cálculo de cada vetor aleatório X_k , no Algoritmo LRS, é dada por

$$\begin{cases} \Lambda_{0_1} = X_0 \\ \Lambda_{k_1} = \xi_k \quad \text{para } k \geq 1. \end{cases} \quad (3.21)$$

Neste caso, tem-se $\ell_k = 1$ para todo k , e $\xi_k \in \mathbb{R}^n$ é um vetor aleatório cuja distribuição condicional dado $\sigma(\mathcal{O}_{k-1})$ é uma distribuição normal multivariada com vetor de média nula e matriz de covariância diagonal definida por,

$$Cov(\xi_k) = \begin{bmatrix} (\lambda_k)_1^2 & 0 & \dots & 0 \\ 0 & (\lambda_k)_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (\lambda_k)_n^2 \end{bmatrix}. \quad (3.22)$$

Observe que o oráculo está associado à coleção dos elementos aleatórios (3.21) gerados até a iteração k .

Função para o cálculo dos iterandos

Agora, vamos definir a função determinística Θ , a qual será utilizada para calcular os iterandos X_k . Para isso, fixada uma iteração k , calcula-se o vetor X_k coordenada a coordenada, como segue

$$(X_k)_i = (X_{k-1}^*)_i + (\xi_k)_i, \quad (3.23)$$

com $i = 1, \dots, n$, onde $(u)_i$ representa a i -ésima coordenada do vetor u . Após o cálculo de todas as coordenadas, se o vetor X_k não estiver no conjunto \mathcal{D} , então é feita uma projeção desse vetor sobre \mathcal{D} . Assim, a função determinística $\Theta : \mathcal{O}_k \rightarrow \mathcal{D}$, utilizada para determinar os iterandos $X_k := \Theta(\mathcal{O}_k)$ é dada por

$$\Theta(\mathcal{O}_k) = \text{proj}_{\mathcal{D}} (X_{k-1}^* + \xi_k), \quad (3.24)$$

onde X_{k-1}^* é a melhor solução após $k-1$ avaliações de função e ξ_k é o vetor aleatório o qual possui distribuição normal multivariada com vetor de média nula e matriz de covariância diagonal dada por (3.22).

Uma vez que ξ_k tem uma distribuição normal multivariada e $Cov(\xi_k)$ é uma matriz diagonal, segue que as variáveis aleatórias $(\xi_k)_1, \dots, (\xi_k)_n$ são condicionalmente independentes dado $\sigma(\mathcal{O}_{k-1})$ e cada $(\xi_k)_i$ tem uma distribuição normal com média nula e desvio padrão $(\lambda_k)_i$.

O algoritmo

Uma vez estabelecidos a coleção de elementos aleatórios e a função determinística, podemos representar o Algoritmo LRS na estrutura do Algoritmo Geral.

Algoritmo 3.2 - Algoritmo LRS [38]

DADOS: $X_0 \in \mathcal{D}$, $X_0^* = X_0$, $\mathcal{O}_0 = \{X_0\}$.

$k = 1$

ENQUANTO O CRITÉRIO DE PARADA NÃO FOR SATISFEITO

Gere $\{\Lambda_{k_1}\} = \{\xi_k\}$

Defina $\mathcal{O}_k := \mathcal{O}_{k-1} \cup \{\Lambda_{k_1}\}$

Calcule $X_k := \Theta(\mathcal{O}_k)$ de acordo com (3.24)

SE $f(X_k) < f(X_{k-1}^*)$

$$X_k^* = X_k$$

SENÃO

$$X_k^* = X_{k-1}^*$$

$k = k + 1$

Convergência em probabilidade

Nesta seção mostra-se que o Algoritmo LRS satisfaz as condições suficientes estabelecidas no Teorema 3.6 as quais garantem a sua convergência em probabilidade. O corolário a seguir é uma releitura de [38, Cor. 3]. Recorde que, $\psi_{\mathcal{D}}(\delta) := \inf_{z \in \mathcal{D}} \mu(B(z, \delta) \cap \mathcal{D})$.

Corolário 3.7 *Sejam \mathcal{D} um conjunto limitado do \mathbb{R}^n tal que $\psi_{\mathcal{D}}(\delta) > 0$, para todo $\delta > 0$, (X_k) e (X_k^*) as sequências geradas pelo Algoritmo LRS. Suponha que f é contínua em um minimizador global x^* de f sobre \mathcal{D} . Além disso, suponha que existe uma subsequência de índices $\{k_j\}_{j \geq 1}$ tal que*

$$\inf_{j \geq 1} \min_{1 \leq i \leq n} (\lambda_{k_j})_i > 0. \quad (3.25)$$

Então, $f(X_k^*) \rightarrow f^*$ q.s.

Demonstração. Por hipótese, temos que existe uma subsequência de índices $\{k_j\}_{j \geq 1}$ tal que (3.25) é satisfeita. E, da construção da sequência no algoritmo,

$$X_{k_j} = \text{proj}_{\mathcal{D}} \left(X_{(k_j)-1}^* + \xi_{k_j} \right),$$

para todo $j \geq 1$, onde ξ_{k_j} é um vetor aleatório cuja distribuição condicional dado $\sigma(\mathcal{O}_{(k_j)-1})$ é a distribuição normal multivariada com vetor de média nula e matriz de covariância dada por

$$\text{Cov}(\xi_{k_j}) = \text{diag} \left((\lambda_{k_j})_1^2, \dots, (\lambda_{k_j})_n^2 \right).$$

Defina $W_j = \xi_{k_j}$ e $V_j = \text{Cov}(W_j)$ para todo $j \geq 1$. Logo,

$$X_{k_j} = \text{proj}_{\mathcal{D}} \left(X_{(k_j)-1}^* + W_j \right),$$

para todo $j \geq 1$. E os autovalores de V_j são as variâncias $(\lambda_{k_j})_1^2, \dots, (\lambda_{k_j})_n^2$ das perturbações aleatórias normais para as diferentes componentes de $X_{(k_j)-1}^*$. Deste modo, o menor autovalor de V_j é $\bar{\lambda}_j := \min_{1 \leq i \leq n} (\lambda_{k_j})_i^2 > 0$. Portanto, o resultado segue do Teorema 3.6. □

3.3.2 O Algoritmo de Busca com Escolha Dinâmica das Coordenadas

Nesta seção descreve-se o Algoritmo de Busca com Escolha Dinâmica das Coordenadas (DDS), proposto por Tolson e Shoemaker [52], nos moldes do Algoritmo Geral.

O Algoritmo DDS foi desenvolvido para resolver problemas de calibração, que surgem no contexto de modelos de simulação de bacias hidrográficas. Estes problemas de calibração tem muitos parâmetros para estimar, que correspondem às variáveis de decisão, e são modelados como problemas de otimização global com restrições do tipo caixa. A formulação matemática destes problemas é do tipo (3.1) onde \mathcal{D} é um hiper-retângulo de \mathbb{R}^n definido por

$$\mathcal{D} = \{x \in \mathbb{R}^n : -\infty < \ell \leq x \leq u < \infty\}, \quad (3.26)$$

onde ℓ e u são os vetores dos limites inferiores e superiores, respectivamente, dos valores das variáveis.

O Algoritmo DDS, em cada iteração gera um ponto tentativo adicionando perturbações aleatórias a algumas ou a todas as coordenadas do melhor ponto corrente X_{k-1}^* . O conjunto de coordenadas a perturbar é determinado probabilisticamente e as perturbações aleatórias são normalmente distribuídas com média nula e um desvio padrão fixo. Numa fase inicial o DDS tem uma busca global, a qual vai tornando-se mais local à medida que o número de avaliações da função (que corresponde ao número de iterações) se aproxima do número máximo de avaliações permitido, o qual é usado também como critério de parada. Essa

transição da busca global para a busca local é alcançada pela redução probabilística e dinâmica do número de coordenadas a perturbar no melhor ponto corrente X_{k-1}^* . Esta é a principal diferença entre os Algoritmos DDS e LRS.

Para colocar o Algoritmo DDS nos moldes do Algoritmo Geral é preciso estabelecer como deve ser gerado o conjunto dos elementos aleatórios $\{\Lambda_{k_j}\}_{1 \leq j \leq \ell_k}$ e a função determinística Θ . A versão original do Algoritmo DDS encontra-se em [52].

Coleção dos elementos aleatórios

O DDS é um algoritmo de busca global baseado numa heurística estocástica de ponto-a-ponto. Dado $r \in (0, 1)$, em cada iteração k , o Algoritmo DDS gera n magnitudes de perturbação dadas por $(\xi_k)_i \sim N(0, \lambda_i^2)$ as quais possuem uma distribuição normal com média nula e desvio padrão λ_i , onde $\lambda_i = r(u_i - \ell_i)$, para todo $i = 1, \dots, n$. Tais magnitudes $(\xi_k)_i$ são utilizadas para calcular o iterando X_k a partir de perturbações sobre o melhor ponto corrente X_{k-1}^* , mas apenas num subconjunto de coordenadas selecionadas aleatoriamente. A probabilidade para selecionar cada coordenada é definida como uma função de k dada por

$$P_k = 1 - \frac{\ln(k)}{\ln(k_{max})},$$

onde k_{max} é o número máximo de iterações. No algoritmo proposto por Tolson e Shoemaker em [52] foi adotado como critério de parada simplesmente o número máximo de avaliações da função $N_{max} > 0$. Note que, $k_{max} = N_{max}$ uma vez que é apenas gerado um único vetor aleatório candidato em cada iteração.

Em cada iteração k , a coleção de elementos aleatórios $\{\Lambda_{k_j}\}_{1 \leq j \leq 4}$ utilizada no cálculo do vetor aleatório X_k , no Algoritmo DDS, é definida por

$$\{\Lambda_{k_1}, \Lambda_{k_2}, \Lambda_{k_3}, \Lambda_{k_4}\} = \{P_k, Z_k, \xi_k, \bar{i}\}, \quad (3.27)$$

onde $P_k \in [0, 1]$ é a probabilidade associada à iteração k , a i -ésima componente $(Z_k)_i$ do vetor Z_k é um número distribuído uniformemente no intervalo $[0, 1]$, $\xi_k = ((\xi_k)_1, \dots, (\xi_k)_n)$, onde $(\xi_k)_i \sim N(0, \lambda_i^2)$, e \bar{i} representa o índice, associado a uma coordenada do melhor ponto corrente, selecionado de forma aleatória.

Deste modo, o oráculo está associado à coleção de tais elementos aleatórios dados em (3.27) e gerados até a iteração k , os quais são utilizados no cálculo de cada iterando de vetor aleatório X_k , para $k = 1, 2, \dots, k_{max}$.

Função para cálculo dos iterandos

Para apresentar o Algoritmo DDS na estrutura do Algoritmo Geral, é preciso definir a função determinística Θ que será utilizada para calcular os iterandos X_k , para $k = 1, 2, \dots, k_{max}$. Uma vez que X_k será obtido perturbando-se algumas coordenadas do melhor ponto corrente até então, X_{k-1}^* , é também preciso definir o conjunto dos índices correspondentes às coordenadas do vetor X_{k-1}^* que serão perturbadas.

Considere a probabilidade P_k e a coleção de elementos aleatórios dada por (3.27). Definimos \bar{J}_k como o conjunto dos índices i tais que o elemento aleatório $(Z_k)_i$ é menor ou igual à probabilidade P_k , isto é,

$$\bar{J}_k = \{i : (Z_k)_i \leq P_k\}. \quad (3.28)$$

Se $(Z_k)_i > P_k$, para todo $i = 1, \dots, n$, então é selecionada aleatoriamente uma coordenada \bar{i} e o cálculo para obter um novo iterando é feito perturbando-se apenas essa coordenada. Assim, definimos o conjunto de índices das coordenadas que serão perturbadas na iteração k por

$$J_k = \begin{cases} \bar{J}_k & \text{se } \bar{J}_k \neq \emptyset \\ \{\bar{i}\} & \text{se } \bar{J}_k = \emptyset. \end{cases} \quad (3.29)$$

Agora, fixada uma iteração k , a função $\Theta_i : \mathcal{O}_k \rightarrow D_i$ dos elementos aleatórios de \mathcal{O}_k utilizada para calcular o iterando aleatório X_k , coordenada a coordenada, é dada por

$$\Theta_i = \begin{cases} \min\{u_i, \max\{(X_{k-1}^*)_i + (\xi_k)_i, \ell_i\}\}, & \text{se } i \in J_k \\ (X_{k-1}^*)_i, & \text{se } i \notin J_k. \end{cases} \quad (3.30)$$

Sendo assim, para cada iteração k , a função determinística $\Theta : \mathcal{O}_k \rightarrow \mathcal{D}$, utilizada para determinar cada iterando $X_k = \Theta(\mathcal{O}_k)$, é definida por

$$\Theta(\mathcal{O}_k) = (\Theta_1, \Theta_2, \dots, \Theta_n), \quad (3.31)$$

com Θ_i definido em (3.30).

O algoritmo

Apresenta-se a seguir o Algoritmo DDS de acordo com a estrutura do Algoritmo Geral.

Algoritmo 3.3 - Algoritmo DDS [52]

DADOS: $X_0 \in \mathcal{D}$, $X_0^* = X_0$, $\mathcal{O}_0 = \{X_0^*\}$.

$k = 1$

ENQUANTO $k \leq k_{max}$

Determine $\{\Lambda_{k_j}\}_{1 \leq j \leq 4} = \{P_k, Z_k, \xi_k, \bar{i}\}$, de acordo com (3.27)

Defina $\mathcal{O}_k := \mathcal{O}_{k-1} \cup \{\Lambda_{k_j}\}_{1 \leq j \leq 4}$

Calcule o conjunto J_k de acordo com (3.29)

Calcule $X_k = \Theta(\mathcal{O}_k)$ de acordo com (3.31)

SE $f(X_k) < f(X_{k-1}^*)$

$$X_k^* = X_k$$

SENÃO

$$X_k^* = X_{k-1}^*$$

$k = k + 1$

Convergência em probabilidade

Esta seção contém uma das contribuições da tese, que consiste na prova de que o Algoritmo DDS satisfaz as condições suficientes estabelecidas no Teorema 3.6 que garantem sua convergência em probabilidade (ou quase certamente).

Primeiro, vamos provar que a sequência gerada pelo Algoritmo DDS admite uma subsequência onde todas as componentes do vetor iterando possuem perturbação aleatória com distribuição normal. Veremos, no lema a seguir, que uma hipótese suficiente para a existência de tal subsequência é que a probabilidade associada em cada iteração $P_k \in (0, 1]$ seja maior que uma constante $\bar{p} > 0$.

Lema 3.8 *Consideremos as sequências (X_k) e (X_k^*) geradas pelo Algoritmo DDS com $P_k \in (0, 1]$ e $P_k \geq \bar{p}$. Existe uma subsequência (X_{k_j}) cujos termos são calculados, coordenada a*

coordenada, por

$$(X_{k_j})_i = \min\{u_i, \max\{(X_{(k_j)-1}^*)_i + (\xi_{k_j})_i, \ell_i\}\}, \quad (3.32)$$

com $(\xi_{k_j})_i \sim N(0, (\lambda_{k_j})_i^2)$. Isto é, todas as componentes de todos os termos da subsequência possuem perturbação aleatória com distribuição normal.

Demonstração. Consideremos o evento sucesso A_k como sendo o evento em que todas as coordenadas do termo X_k , da sequência gerada pelo DDS, possuem perturbação aleatória com distribuição normal. Em outras palavras, o evento A_k está associado ao termo $X_k = ((X_k)_1, \dots, (X_k)_n)$ obtidos perturbando-se todas as coordenadas do melhor ponto corrente até então, X_{k-1}^* . Neste caso, temos que $i \in \bar{J}_k$ para todo $i = 1, \dots, n$, com \bar{J}_k dado em (3.28).

Dessa forma, temos

$$[A_k] = [(Z_k)_i \leq P_k, \quad \text{para todo } i = 1, \dots, n],$$

onde $(Z_k)_i \sim U[0, 1]$.

Queremos mostrar que o evento A_k ocorre infinitas vezes, isto é, que

$$P(A_k \text{ infinitas vezes}) = 1.$$

Uma vez que os eventos $[(Z_k)_1 \leq P_k], \dots, [(Z_k)_n \leq P_k]$ são independentes, temos

$$P(A_k) = \prod_{i=1}^n P((Z_k)_i \leq P_k). \quad (3.33)$$

Como cada $(Z_k)_i \sim U[0, 1]$, temos que a função de densidade é dada por

$$g(z) = \begin{cases} 1, & z \in [0, 1] \\ 0, & z \notin [0, 1]. \end{cases}$$

Assim, para cada i , temos

$$P((Z_k)_i \leq P_k) = \int_{-\infty}^{P_k} g(z) dz = \int_{-\infty}^0 0 dz + \int_0^{P_k} 1 dz = P_k.$$

Substituindo em (3.33), tem-se

$$P(A_k) = \prod_{i=1}^n P_k = P_k^n.$$

Por hipótese, temos que $P_k \geq \bar{p}$, logo

$$\sum_{k=1}^{\infty} P(A_k) = \sum_{k=1}^{\infty} P_k^n \geq \sum_{k=1}^{\infty} \bar{p}^n = \bar{p}^n \sum_{k=1}^{\infty} 1 = \infty.$$

Portanto, pelo item (ii) do Lema 2.31 de Borel-Cantelli segue que

$$P(A_k \text{ infinitas vezes}) = 1.$$

Isto significa que o evento sucesso A_k ocorre infinitas vezes. Dessa forma, de acordo com a construção da sequência no Algoritmo DDS, (X_k) admite uma subsequência (X_{k_j}) tal que todas as componentes de seus termos possuem perturbação aleatória com distribuição normal. Ou seja, os termos da subsequência (X_{k_j}) são calculados, coordenada a coordenada, por

$$(X_{k_j})_i = \min\{u_i, \max\{(X_{(k_j)-1}^*)_i + (\xi_{k_j})_i, \ell_i\}\}, \quad (3.34)$$

com $(\xi_{k_j})_i \sim N(0, (\lambda_{k_j})_i^2)$, para $i = 1, \dots, n$.

□

Uma vez provada a existência de tal subsequência, através do Lema 3.8, podemos dar continuidade à prova da convergência em probabilidade do Algoritmo DDS através do teorema a seguir.

Teorema 3.9 *Sejam \mathcal{D} um conjunto limitado tal que $\psi_{\mathcal{D}}(\delta) > 0$, para todo $\delta > 0$, e suponha que f é contínua em um minimizador global x^* de f sobre \mathcal{D} . Se (X_k) e (X_k^*) são as sequências gerada pelo Algoritmo DDS com $P_k \in (0, 1]$ e $P_k \geq \bar{p}$, então $f(X_k^*) \rightarrow f^*$ q.s.*

Demonstração. Pelo Lema 3.8 existe uma subsequência (X_{k_j}) dada por (3.32), com $(\xi_{k_j})_i \sim N(0, (\lambda_{k_j})_i^2)$ e $(\lambda_{k_j})_i = (\lambda)_i = r(u_i - \ell_i)$. Deste modo, $\xi_{k_j} \sim N(0, V_j)$ é um vetor aleatório cuja distribuição condicional dado $\sigma(\mathcal{O}_{(k_j)-1})$ é a distribuição normal multivariada com vetor de média nula e matriz de covariância V_j dada por

$$V_j = \text{Cov}(\xi_{k_j}) = \text{diag}((\lambda)_1^2, \dots, (\lambda)_n^2).$$

Neste caso, temos que os autovalores de V_j são as variâncias $(\lambda)_1^2, \dots, (\lambda)_n^2$ das perturbações aleatórias normais para as diferentes componentes de $X_{(k_j)-1}^*$. Assim, o menor autovalor de V_j é dado por

$$\bar{\lambda}_j := \min_{1 \leq i \leq n} (\lambda)_i^2 > 0.$$

Portanto, o resultado segue do Teorema 3.6.

□

3.3.3 O Algoritmo de Busca com Escolha Dinâmica das Coordenadas usando Modelos de Superfície de Resposta

Nesta seção vamos descrever o Algoritmo de Busca com Escolha Dinâmica das Coordenadas usando Modelos de Superfície de Resposta (do inglês, *DYnamic COordinate search usign Response Surface models* - DYCORS), desenvolvido por Regis e Shoemaker em [42]. Em cada iteração k , o DYCORS constrói e mantém um modelo de superfície de resposta da função objetivo e usa uma estratégia de escolha dinâmica das coordenadas para gerar múltiplos pontos tentativos. Destes múltiplos pontos tentativos, será selecionado o melhor ponto para ser avaliado em f . O Algoritmo DYCORS foca na resolução de problemas de otimização de grande dimensão, em que as avaliações da função objetivo são de custo computacional elevado. O que motivou este estudo foi o interesse em usar múltiplos pontos tentativos sem aumentar o número de avaliações da função objetivo na resolução dos problemas de otimização com restrições gerais, o qual será estudado no Capítulo 4.

Tal como o DDS, o Algoritmo DYCORS foi desenvolvido para resolver problemas de otimização global com restrições do tipo caixa, isto é, para problemas do tipo (3.1) onde \mathcal{D} é dado por (3.26). No entanto, a principal diferença entre os Algoritmos DYCORS e DDS, é que em cada iteração o DYCORS gera n_t pontos tentativos enquanto que o DDS gera apenas um. Devido aos múltiplos pontos tentativos gerados em cada iteração, utilizamos a notação X_k^t para designar o t -ésimo ponto tentativo gerado na iteração k .

Para que a notação fique uniforme em todo o texto, o Algoritmo DYCORS será apresentado nos moldes do Algoritmo Geral. Para tal, será estabelecido como devem ser gerados o

conjunto dos elementos aleatórios Λ_{k_j} e a função determinística Θ . Para uma leitura mais detalhada do Algoritmo DYCORS, recomenda-se a consulta de [42].

Pontos Iniciais

No Algoritmo DYCORS, inicialmente é gerado um conjunto de $2(n + 1)$ pontos iniciais aleatoriamente no espaço de busca \mathcal{D} , do seguinte modo:

$$I = \{(X_j) \in \mathcal{D} : X_j = \ell + \beta_j(u - \ell), \quad \text{com} \quad 1 \leq j \leq 2(n + 1)\}. \quad (3.35)$$

Neste caso, cada vetor $\beta_j \in [0, 1]^n$ é gerado usando um hipercubo latino simétrico, (do inglês, *symmetric Latin hypercube designs* - SLHDs), proposto em [53].

Coleção de elementos aleatórios

Dado $r \in (0, 1]$, em cada iteração k , o Algoritmo DYCORS gera os vetores ξ_k^t , com $t = 1, \dots, n_t$, onde n_t é o número de pontos tentativos (vetores aleatórios gerados a cada iteração). As componentes de ξ_k^t são as magnitudes de perturbação para as respectivas componentes do ponto tentativo X_k^t , com $t = 1, \dots, n_t$. Estas magnitudes de perturbação possuem distribuição normal com média nula e desvio padrão $(\lambda_{k-1})_i$, isto é, $(\xi_k^t)_i \sim N(0, (\lambda_{k-1})_i^2)$. No DYCORS é feito um ajuste do tamanho do passo através do controle do desvio padrão associado ao tamanho da magnitude de perturbação. Este parâmetro é inicializado por $(\lambda_0)_i = r(u_i - \ell_i)$, para todo $i = 1, \dots, n$. Ao longo do processo iterativo seu valor é ajustado conforme a iteração seja considerada de sucesso ou insucesso, usando-se os contadores de sucesso c_{suc} , de falha c_{falha} e os parâmetros tolerância máxima de falhas ou insucessos, τ_{falha} , e tolerância máxima de sucessos, τ_{suc} . No entanto, no DYCORS uma iteração é considerada de sucesso se o melhor ponto tentativo provocar um decréscimo suficiente na função objetivo, isto é, se $(f(X_{k-1}^*) - f(X_k^{t*})) > 10^{-3}|f(X_{k-1}^*)|$, onde X_{k-1}^* é o melhor ponto corrente e X_k^{t*} o melhor ponto tentativo. A iteração é considerada de insucesso se $(f(X_{k-1}^*) - f(X_k^{t*})) \leq 10^{-3}|f(X_{k-1}^*)|$ ou se o melhor ponto corrente não for atualizado.

As magnitudes $(\xi_k^t)_i$ são utilizadas para calcular os n_t pontos tentativos X_k^t , com $t = 1, \dots, n_t$, a partir de perturbações sobre o melhor ponto corrente X_{k-1}^* , nas coordenadas selecionadas aleatoriamente.

A coleção de elementos aleatórios $\{\Lambda_{k_j}\}_{1 \leq j \leq 3n_t+1}$ utilizada no cálculo dos n_t vetores aleatórios X_k^t , com $t = 1, \dots, n_t$, no Algoritmo DYCORS, é dada por

$$\{\Lambda_{k_j}\}_{1 \leq j \leq 3n_t+1} = \{P_k, Z_k^t, \xi_k^t, \bar{i}^t\}, \quad (3.36)$$

onde $P_k \in [0, 1]$ é a probabilidade calculada em cada iteração e \bar{i}^t representa o índice de uma coordenada do melhor ponto corrente, selecionado de forma aleatória para cada ponto tentativo. Além disso, para cada $t = 1, \dots, n_t$, temos que $Z_k^t \in \mathbb{R}^n$ é um vetor cuja i -ésima componente $(Z_k^t)_i$ é um número distribuído uniformemente no intervalo $[0, 1]$ e $\xi_k^t = ((\xi_k^t)_1, \dots, (\xi_k^t)_n)$, onde $(\xi_k^t)_i \sim N(0, (\lambda_{k-1})_i^2)$.

Deste modo, o oráculo está associado à coleção de tais elementos aleatórios dados em (3.36) e gerados até a iteração k , os quais são utilizados no cálculo dos n_t pontos tentativos aleatórios X_k^t .

Cálculo da probabilidade

Tal como no Algoritmo DDS, é usada uma probabilidade na seleção das coordenadas a perturbar. No Algoritmo DYCORS, em cada iteração k , é calculada a probabilidade P_k a qual é uma função do número de avaliações da função objetivo, e definida por

$$P_k = \left(1 - \frac{\ln(k - n_0 + 1)}{\ln(N_{max} - n_0)}\right) \min \left\{ \frac{20}{n}, 1 \right\}, \quad (3.37)$$

onde n é a dimensão do problema e N_{max} é o número máximo de avaliações da função objetivo permitido.

Função para o cálculo dos múltiplos iterandos de vetores aleatórios

Agora definiremos a função determinística utilizada para calcular cada um dos pontos tentativos X_k^t , com $t = 1, 2, \dots, n_t$. De acordo com as ideias do DDS, o DYCORS gera seus

pontos tentativos perturbando todas ou algumas coordenadas do melhor ponto corrente até o momento, X_{k-1}^* . Deste modo é necessário definir o conjunto dos índices correspondentes às componentes do vetor X_{k-1}^* que serão perturbadas.

Considere a probabilidade P_k e a coleção de elementos aleatórios dada em (3.36). Para cada $t = 1, \dots, n_t$, definimos \bar{J}_k^t como o conjunto dos índices i 's tais que o elemento aleatório $(Z_k^t)_i$ é menor ou igual à probabilidade P_k . Ou seja,

$$\bar{J}_k^t = \{i : (Z_k^t)_i \leq P_k\}.$$

Se $(Z_k^t)_i > P_k$, para todo $i = 1, \dots, n$, então uma coordenada \bar{i} é selecionada aleatoriamente. Em seguida, o cálculo para obter um novo iterando é feito perturbando apenas a coordenada \bar{i} . Assim, para cada $t = 1, \dots, n_t$, definimos o conjunto de índices das coordenadas que serão perturbadas na iteração k , como segue

$$J_k^t = \begin{cases} \bar{J}_k^t & \text{se } \bar{J}_k^t \neq \emptyset \\ \{\bar{i}\} & \text{se } \bar{J}_k^t = \emptyset. \end{cases} \quad (3.38)$$

Agora, fixada uma iteração k , a função $\Theta_i^t : \mathcal{O}_k \rightarrow D_i$ dos elementos aleatórios de \mathcal{O}_k utilizada para calcular cada um dos múltiplos iterandos aleatórios, coordenada a coordenada, é dada por

$$\Theta_i^t = \begin{cases} \min\{u_i, \max\{(X_{k-1}^*)_i + (\xi_k^t)_i, \ell_i\}\}, & \text{se } i \in J_k^t \\ (X_{k-1}^*)_i, & \text{se } i \notin J_k^t. \end{cases} \quad (3.39)$$

Sendo assim, para cada iteração k , a função determinística $\Theta^t : \mathcal{O}_k \rightarrow \mathcal{D}$, utilizada para determinar cada um dos múltiplos iterandos $X_k^t = \Theta^t(\mathcal{O}_k)$, é definida por

$$\Theta^t(\mathcal{O}_k) = (\Theta_1^t, \Theta_2^t, \dots, \Theta_n^t). \quad (3.40)$$

Interpolação através de funções de base radial

A grande diferença entre os Algoritmos DDS e DYCORS é que, em cada iteração, o DYCORS gera um número n_t de pontos tentativos enquanto que o DDS gera um único ponto tentativo. No DYCORS o melhor ponto tentativo é selecionado utilizando modelos de

superfície de resposta cuja aproximação da função objetivo é feita via funções de base radial. Assim, esta estratégia permite trabalhar com múltiplos pontos tentativos a reduzido custo computacional, sendo apenas avaliado em f o melhor destes pontos. O modelo RBF utilizado aqui é baseado em [42].

Sejam $X_1, \dots, X_m \in \mathbb{R}^n$, m pontos amostrais distintos com avaliações da função objetivo conhecidas $f(X_1), \dots, f(X_m)$. O modelo RBF aproxima a função objetivo através da seguinte função de interpolação,

$$S(X) = \sum_{i=1}^k \eta_i \phi(\|X - X_i\|) + p(X), \quad (3.41)$$

onde $X \in \mathbb{R}^n$, $\eta_i \in \mathbb{R}$, para $i = 1, \dots, m$, $\|\cdot\|$ é a norma Euclidiana, p é um polinômio linear de n variáveis e a função ϕ tem a forma cúbica, $\phi(y) = y^3$, sugerida por [42]. No entanto, outras funções podem ser usadas como, por exemplo, $\phi(y) = y^2 \ln(y)$ ou a Gaussiana $\phi(y) = e^{-\gamma y^2}$, com $\gamma > 0$ [36].

Para obter a aproximação da função objetivo, avaliamos a função de interpolação S nos m pontos amostrais distintos. Para isso, considere $\Phi \in \mathbb{R}^{m \times m}$ a matriz definida por $\Phi := \phi(\|X_i - X_j\|)$, $i, j = 1, \dots, m$, e $M_P \in \mathbb{R}^{k \times (n+1)}$ a matriz cuja i -ésima linha é dada por $[1 \ X_i^T]$. Deste modo, o modelo RBF cúbico que interpola os pontos $(X_1, f(X_1)), \dots, (X_m, f(X_m))$ é obtido resolvendo o seguinte sistema,

$$\begin{pmatrix} \Phi & M_P \\ M_P^T & 0 \end{pmatrix} \begin{pmatrix} \eta \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0_{n+1} \end{pmatrix}, \quad (3.42)$$

onde $F = (f(X_1), \dots, f(X_m))$, $\eta = (\eta_1, \dots, \eta_m)^T \in \mathbb{R}^m$ e $c = (c_1, \dots, c_{n+1}) \in \mathbb{R}^{n+1}$ são os coeficientes do polinômio linear p .

Seleção do melhor ponto tentativo

No Algoritmo DYCORDS o melhor ponto aleatório tentativo, denotado por X_k^{t*} , é selecionado como base no valor estimado pelo modelo de aproximação S , designado por critério RBF, e na distância mínima dos pontos previamente avaliados, designada por critério de

distância. É definida uma pontuação ponderada destes dois critérios dado que é desejável procurar um ponto tentativo com um menor valor de S e que esteja longe dos pontos previamente avaliados, a fim de melhorar o modelo RBF corrente S .

Para isso, seguindo as ideias de [42], é utilizado o peso padrão para o critério RBF como sendo $\Gamma = \{v_1, \dots, v_s\}$, com $0 \leq v_1 \leq \dots \leq v_s \leq 1$. Considerando k o número de avaliações da função objetivo e n_0 o número de pontos necessários para inicializar o modelo RBF, temos que o peso w_k^R para o critério RBF, para todo $k \geq n_0$, é dado por

$$w_k^R = \begin{cases} v_{\text{mod}(k-n_0+1,s)} & \text{se } \text{mod}(k-n_0+1, s) \neq 0 \\ v_s & \text{caso contrário.} \end{cases}$$

O peso para o critério de distância é simplesmente $1 - w_k^R$. A seguir, apresentamos o Algoritmo DYCORS adaptado para a notação do Algoritmo Geral. O Algoritmo DYCORS, na sua forma original, encontra-se com mais detalhes em [42].

O Algoritmo

Algoritmo 3.4 - Algoritmo DYCORS [42]

DADOS: $n, n_t, u, \ell, k_{\max}, \lambda_0, \tau_{\text{suc}}, \tau_{\text{falha}}, \gamma_{\text{tol}}, \Gamma, c_{\text{suc}} = 0, c_{\text{falha}} = 0$.

Pontos iniciais: $I = \{X_1, \dots, X_{n_0}\} \subset \mathcal{D}$, selecione o ponto com o menor valor de f, X_0^* , e defina $\mathcal{O}_0 = \{X_0^*\}$.

Faça $k = 1, n_f = n_0, \mathcal{A}_k = I, X_{k-1}^* = X_0^*, \mathcal{O}_{k-1} = \mathcal{O}_0$ e $\lambda_{k-1} = \lambda_0$.

ENQUANTO $k \leq k_{\max}$

Ajuste ou atualize S usando os pontos:

$$\mathcal{B}_k = \{(X, f(X)) : X \in \mathcal{A}_k\} = \{(X_i, f(X_i)) : i = 1, \dots, k\}$$

Gere $\Lambda_{k_j} = \{P_k, Z_k^t, \xi_k^t, \bar{i}^t\}$, com $t = 1, \dots, n_t$ e defina $\mathcal{O}_k := \mathcal{O}_{k-1} \cup \Lambda_{k_j}$

Calcule o conjunto J_k^t , com $t = 1, \dots, n_t$, de acordo com (3.38)

Calcule $\Upsilon_k = \{X_k^t, t = 1, \dots, n_t\}$, onde $X_k^t := \Theta^t(\mathcal{O}_k)$ de acordo com (3.40)

Selecione o melhor ponto tentativo: $X_k^{t*} = \text{RBF}_D(\Upsilon_k, \mathcal{B}_k, S)$

Calcule $f(X_k^{t*})$ e faça $n_f = n_f + 1$

SE $f(X_k^{t*}) < f(X_{k-1}^*)$

$$X_k^* = X_k^{t*}$$

SE $(f(X_{k-1}^*) - f(X_k^{t*})) > 10^{-3}|f(X_{k-1}^*)|$

$$c_{suc} = c_{suc} + 1 \text{ e } c_{falha} = 0$$

SENÃO

$$c_{falha} = c_{falha} + 1 \text{ e } c_{suc} = 0$$

SENÃO

$$X_k^* = X_{k-1}^*$$

$$c_{falha} = c_{falha} + 1 \text{ e } c_{suc} = 0$$

Ajuste do tamanho do passo:

SE $c_{falha} \geq \tau_{falha}$

$$\lambda_k = \lambda_{k-1}/2 \text{ e } c_{falha} = 0$$

SE $c_{suc} \geq \tau_{suc}$

$$\lambda_k = 2\lambda_{k-1} \text{ e } c_{suc} = 0$$

Atualização do conjunto dos pontos amostrais: $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{X_k^{t*}\}$

$k = k + 1$.

Apresenta-se a seguir, a descrição do Algoritmo interno RBF_D , utilizado na seleção do melhor ponto tentativo do DYCORS.

Algoritmo 3.5 - *Algoritmo RBF_D [42]*

DADOS: $\Upsilon_k, \mathcal{B}_k, S$

Para cada $Y \in \Upsilon_k$ calcule a pontuação $V_k^R(Y)$ entre 0 e 1 para o critério RBF:

$$S(Y), S^{max} = \max_{Y \in \Upsilon_k} S(Y) \text{ e } S^{min} = \min_{Y \in \Upsilon_k} S(Y).$$

SE $S^{max} - S^{min} > \gamma_{tol}$

$$V_k^R(Y) = \frac{S(Y) - S^{min}}{S^{max} - S^{min}}$$

SENÃO

$$V_k^R(Y) = 1$$

Para cada $Y \in \Upsilon_k$ calcule a pontuação $\Delta_k(Y)$ entre 0 e 1 para o critério distância:

$$\Delta_k(Y) = \min_{1 \leq i \leq k} \|Y - Y_i\|, \Delta_k^{max} = \max_{Y \in \Upsilon_k} \Delta_k(Y) \text{ e}$$

$$\Delta_k^{min} = \min_{Y \in \Upsilon_k} \Delta_k(Y).$$

$$\text{SE } \Delta_k^{max} - \Delta_k^{min} > \gamma_{tol}$$

$$V_k^D(Y) = \frac{\Delta_k^{max} - \Delta_k(Y)}{\Delta_k^{max} - \Delta_k^{min}}$$

SENÃO

$$V_k^D(Y) = 1$$

Para cada $Y \in \Upsilon_k$ calcule a pontuação ponderada:

$$\mathcal{W}_k(Y) = w_k^R V_k^R(Y) + (1 - w_k^R) V_k^D(Y)$$

Selecione o melhor ponto tentativo para avaliação em f : $X_k^{t*} = \underset{Y \in \Upsilon_k}{\operatorname{argmin}} \mathcal{W}_k(Y)$.

Convergência em probabilidade

Os iterandos do Algoritmo DYCORS são gerados da mesma maneira que os iterandos do Algoritmo DDS, as únicas diferenças são o número de pontos tentativos gerados e a maneira como é selecionado o melhor ponto tentativo. Dessa forma, temos que a sequência (X_k) gerada pelo DYCORS admite uma subsequência cujos termos possuem todas as componentes com perturbação aleatória seguindo uma distribuição normal. Portanto, a demonstração da convergência em probabilidade do DYCORS segue de forma análoga à convergência em probabilidade do DDS, isto é, de acordo com o Lema 3.8 e o Teorema 3.9 apresentados na Seção 3.3.2.

Capítulo 4

Algoritmos estocásticos com controle de inviabilidade

Este capítulo contém algumas das contribuições mais significativas da tese. O objetivo é propor um algoritmo estocástico para resolver problemas de otimização com restrições gerais, em que é necessário controlar a inviabilidade, e fazer um estudo de condições suficientes para sua convergência quase certamente. Para tratar a inviabilidade das restrições gerais envolvidas no problema, recorre-se a um método de filtro [35].

Primeiro propõe-se um algoritmo geral estocástico de filtro, onde os iterandos são vetores aleatórios cujas realizações são geradas de acordo com alguma distribuição de probabilidade. Para o algoritmo geral estocástico de filtro é efetuado um estudo de condições suficientes para que este seja convergente. Com base neste estudo são propostos dois algoritmos estocásticos de filtro práticos, os quais baseiam-se na ideia de busca com escolha dinâmica das coordenadas do DDS, estudado no Capítulo 3. Começa-se por propor e desenvolver um algoritmo nos moldes do algoritmo geral estocástico de filtro, denominado Algoritmo de Busca com Escolha Dinâmica das Coordenadas e de Filtro (FDDS), e que usa a ideia de escolha dinâmica das coordenadas do DDS. Em seguida, propõe-se e desenvolve-se o Algoritmo de Busca com Escolha Dinâmica das Coordenadas e de Filtro usando interpolação por Funções de Base Radial (FDDSRBF). É de referir que o algo-

ritmo FDDSRBF proposto combina o algoritmo FDDS com um modelo de interpolação da função objetivo por funções de base radial, para ultrapassar a dificuldade do elevado custo computacional associado as avaliações da função objetivo, e foi inspirado no estudo do Algoritmo DYCORS apresentado no Capítulo 3. O FDDSRBF apresenta uma redução bastante significativa no número de avaliações da função objetivo comparativamente ao FDDS. Neste capítulo, para cada um dos três algoritmos apresentar-se-á a sua descrição detalhada e o respectivo estudo acerca das condições suficientes de convergência.

4.1 O problema

Neste capítulo está-se interessado em resolver problemas de otimização com restrições gerais do tipo (1.1) em que o conjunto viável \mathcal{D}_V é definido por

$$\mathcal{D}_V = \{x \in \mathcal{D} : c_{\mathcal{I}}(x) \leq 0, c_{\mathcal{E}}(x) = 0\}$$

onde $c_{\mathcal{I}}$, $c_{\mathcal{E}}$ são os vetores das funções de restrição de desigualdade e igualdade, respectivamente; \mathcal{I} e \mathcal{E} são dois conjuntos de índices finitos, e \mathcal{D} é dado por (3.26). Assim, a formulação geral para estes problemas é

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeito a} && c_{\mathcal{I}}(x) \leq 0 \\ &&& c_{\mathcal{E}}(x) = 0 \\ &&& x \in \mathcal{D} \end{aligned} \tag{4.1}$$

onde $f : A \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \mathcal{I} \cup \mathcal{E}$ e tal que $A \supseteq \mathcal{D} \supseteq \mathcal{D}_V$. Assumimos que a função f é limitada inferiormente em \mathcal{D}_V , ou seja, $f^* := \inf_{x \in \mathcal{D}_V} f(x) > -\infty$. Recordar-se que x^* denota um minimizador global de f sobre o conjunto \mathcal{D}_V , isto é, $f(x^*) \leq f(x)$ para todo $x \in \mathcal{D}_V$. Se x^* existe, temos $f^* = f(x^*)$.

Para resolver problemas do tipo (4.1) é necessário um controle da inviabilidade, o que foi feito através da estratégia dos métodos de filtro, sobre os quais falaremos brevemente na próxima seção.

4.2 Métodos de filtro

Nesta seção apresenta-se uma rápida descrição sobre *métodos de filtro* introduzidos por Fletcher e Leyffer em [15] para controlar o progresso de um algoritmo de otimização que resolve problemas gerais de programação não linear. Estudos mais detalhados sobre método de filtro, sua convergência e aplicações podem ser encontrados em [7, 8, 13, 14, 16, 17, 19, 26, 28, 35, 45, 46].

Ao utilizar determinado método de otimização, para problemas com controle de inviabilidade, é necessário que os pontos candidatos obtidos sejam submetidos a algum critério de aceitação, ou seja, é preciso incorporar uma metodologia que avalie o ponto candidato calculado, de modo que este seja aceito apenas se for “razoavelmente” melhor comparativamente com o ponto corrente. No contexto de um método de filtro, é necessário definir uma função para medir a inviabilidade num certo ponto x .

Considerar-se-á a função medida de inviabilidade $h : \mathbb{R}^n \rightarrow \mathbb{R}_+$ definida por

$$h(x) = \|c^+(x)\|, \quad (4.2)$$

onde $\|\cdot\|$ é uma norma arbitrária e $c^+ : \mathbb{R}^n \rightarrow \mathbb{R}^m$ é uma função definida da seguinte maneira,

$$c_i^+(x) = \begin{cases} c_i(x), & \text{se } i \in \mathcal{E} \\ \max\{0, c_i(x)\}, & \text{se } i \in \mathcal{I}. \end{cases}$$

Note que, para $x \in \mathcal{D}$ temos que $h(x) = 0$ se, e somente se, x é um ponto viável, isto é, $x \in \mathcal{D}_V$; caso contrário $h(x)$ é positiva.

O método de filtro reformula o problema (4.1) como um problema de otimização bi-objetivo minimizando a função objetivo f bem como a função de medida da inviabilidade h , em \mathcal{D} , como segue

$$\min_{x \in \mathcal{D}} (f(x), h(x)). \quad (4.3)$$

Estes métodos baseiam-se no conceito de dominância, da otimização multi-objetivo, onde se constrói um filtro que aceita iterandos apenas se eles melhoram o valor da função objetivo ou o valor da função de medida da inviabilidade, baseado na seguinte *Regra de dominância de Pareto*.

Definição 4.1 Diz-se que um ponto x domina um ponto x^+ , ou equivalentemente, o par $(f(x), h(x))$ domina o par $(f(x^+), h(x^+))$, se, e somente se, $f(x) \leq f(x^+)$ e $h(x) \leq h(x^+)$ e em que pelo menos uma das desigualdades é estrita.

A Figura 4.1 ilustra a definição de dominância, onde a área pintada representa a região dos pontos dominados por x no plano $f \times h$.

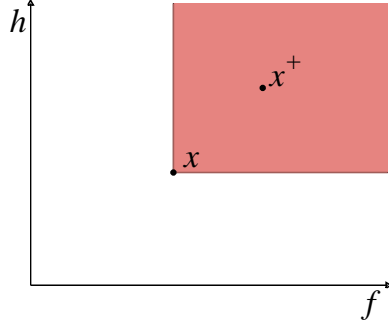


Figura 4.1: Regra de dominância de Pareto

Os algoritmos de filtro consideram, em cada iteração k , um filtro F_k que é um conjunto de pares $(f(x_\ell), h(x_\ell)) \in \mathbb{R}^2$, que correspondem a uma coleção de pontos x_ℓ , tais que nenhum ponto (ou par) domina nenhum outro do filtro. Diz-se que o ponto x_ℓ (ou o par que lhe corresponde) é aceito pelo filtro se não for dominado por nenhum ponto (ou par) do filtro. É de salientar que quando um ponto é adicionado ao filtro, todos os pontos (ou pares) que ficam dominado por ele são removidos do filtro. Para simplificar a notação, utilizamos (f_ℓ, h_ℓ) para denotar o par $(f(x_\ell), h(x_\ell))$.

Associado ao filtro está o conceito de envelope (ou margem). Para evitar a convergência para pontos de acumulação que são inviáveis, isto é, para pontos x^* tais que $h(x^*) > 0$, é adicionado uma margem em torno do filtro corrente. Para tal, dada uma constante $\alpha \in (0, 1)$, um par (f_j, h_j) do filtro F_k define uma *região proibida* $\mathcal{R}_j \subset \mathbb{R}^n$ que pode ser definida por

$$\mathcal{R}_j = \{x \in \mathbb{R}^n \mid f(x) \geq f_j - \alpha h_j \quad \text{e} \quad h(x) \geq (1 - \alpha)h_j\}, \quad (4.4)$$

como proposto originalmente em [15], ou por

$$\mathcal{R}_j = \{x \in \mathbb{R}^n \mid f(x) + \alpha h(x) \geq f_j \quad \text{e} \quad h(x) \geq (1 - \alpha)h_j\}, \quad (4.5)$$

como proposto em [7].

Um filtro baseado na regra (4.4) será chamado de *filtro reto* e o baseado em (4.5) será chamado de *filtro inclinado*. Embora, resultados mais fortes de convergência global tenham sido provados para algoritmos baseados no filtro inclinado, experiências computacionais não mostraram diferenças numéricas significativas entre ambas as regras de filtro, como discutido em [35].

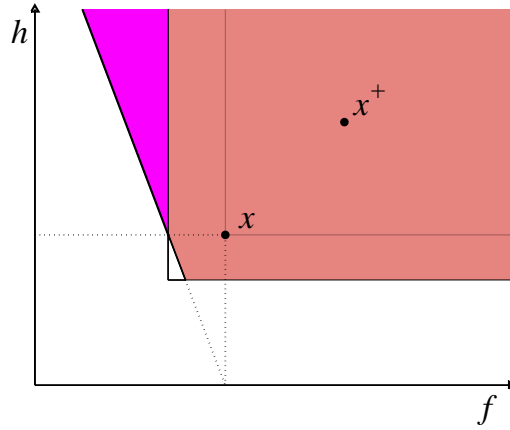


Figura 4.2: Diferença entre uma região proibida com filtro reto e inclinado

A Figura 4.2 mostra as diferenças entre uma região proibida pelo ponto x , ou equivalentemente pelo par $(f(x), h(x))$, utilizando um filtro reto e utilizando um filtro inclinado, respectivamente. Note que, a região proibida utilizando o filtro inclinado abrange uma área maior do que a região proibida utilizando o filtro reto. A área da região inclinada inclui uma região triangular à esquerda na Figura 4.2 e exclui o triângulo inferior que pertence à região proibida utilizando o filtro reto.

Assim, um ponto x , ou equivalentemente o par $(f(x), h(x))$, é dito ser *não dominado* pelo par (f_j, h_j) se $x \notin \mathcal{R}_j$, com \mathcal{R}_j dado por (4.4) ou (4.5). Portanto, diz-se que x , ou equivalentemente o par $(f(x), h(x))$, é *aceito pelo filtro* F_k se ele é não dominado por nenhum par do filtro. Isto é, se para todo $(f_\ell, h_\ell) \in F_k$, tem-se

$$f(x) < f_\ell - \alpha h_\ell \quad \text{ou} \quad h(x) < (1 - \alpha)h_\ell \quad (4.6)$$

para o caso do filtro reto, e

$$f(x) < f_\ell - \alpha h(x) \quad \text{ou} \quad h(x) < (1 - \alpha)h_\ell, \quad (4.7)$$

para o caso do filtro inclinado. A Figura 4.3 mostra as regiões em \mathbb{R}^2 formadas pelos pares $(f(x), h(x))$ dominados pelo par (f_j, h_j) para as regras de filtro reto e inclinado, dadas em (4.4) e (4.5) respectivamente.

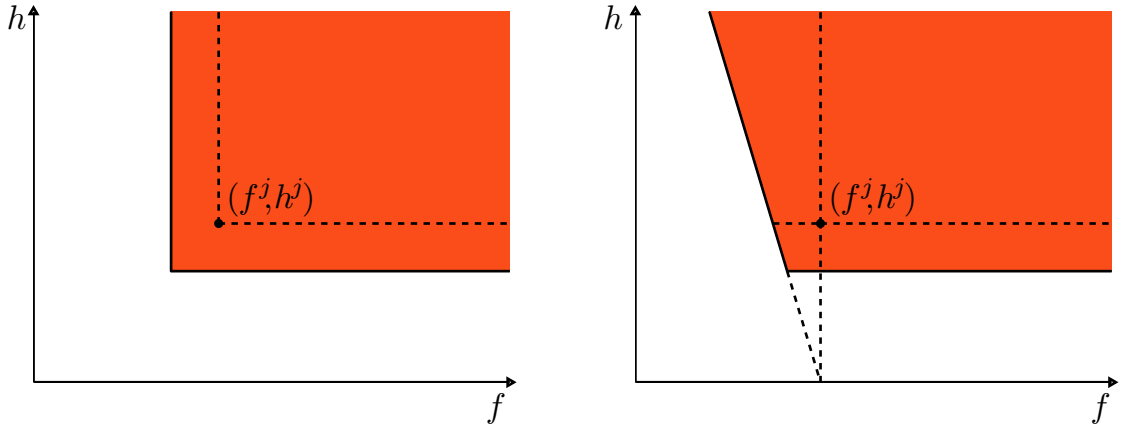


Figura 4.3: Filtros reto e inclinado

Vamos usar uma adaptação de um algoritmo geral de filtro, apresentado em [44], que permite uma grande liberdade no cálculo do passo e na escolha do critério de filtro, reto ou inclinado. Tal algoritmo constrói uma sequência de conjuntos F_0, F_1, \dots, F_{k-1} , formados pelos pares (f_ℓ, h_ℓ) , onde F_{k-1} é denominado filtro corrente. O algoritmo também define o conjunto \mathcal{F}_{k-1} como sendo a região corrente permanentemente proibida em \mathbb{R}^n e uma região corrente temporariamente proibida dada por $\bar{\mathcal{F}}_{k-1} = \mathcal{F}_{k-1} \cup \mathcal{R}_{k-1}$.

A Figura 4.4 apresenta o filtro permanente, representado pelo conjunto

$$F_k = \{(f_i, h_i), (f_j, h_j), (f_\ell, h_\ell)\},$$

e o filtro temporário dado por $\bar{F}_k = F_k \cup \{(f_k, h_k)\}$, para o filtro reto e inclinado. As regiões pintadas são formadas pelos pares $(f(x), h(x))$ tais que $x \in \bar{\mathcal{F}}_k$.

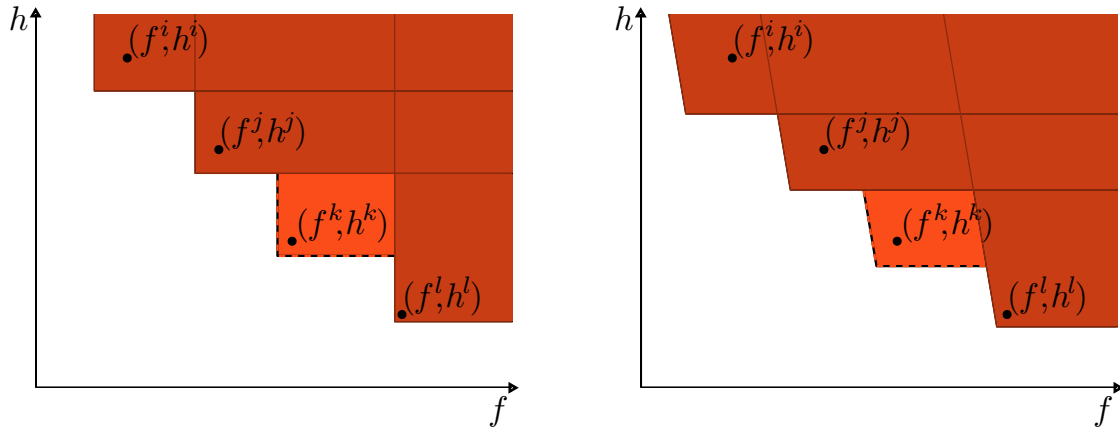


Figura 4.4: Regiões proibidas no plano $f \times h$ para o filtro reto e inclinado

4.3 Um algoritmo geral estocástico de filtro

Nesta seção propõe-se um algoritmo geral estocástico de filtro para resolver problemas de otimização com restrições gerais do tipo (4.1). O controle da inviabilidade das restrições gerais é feito recorrendo ao método de filtro, descrito na Seção 4.2, o qual reformula o problema (4.1) como o problema de otimização bi-objetivo dado por (4.3). A seguir, faz-se um estudo da convergência, no sentido probabilístico, do algoritmo geral estocástico de filtro.

4.3.1 Algoritmo geral

No algoritmo geral estocástico de filtro aqui proposto, os iterandos são gerados como uma sequência de vetores aleatórios $(X_k)_{k \geq 1}$ definidos num espaço de probabilidade, (Ω, \mathcal{F}, P) , onde Ω é o conjunto de resultados, \mathcal{F} é uma σ -álgebra e $P : \mathcal{F} \rightarrow [0, 1]$ é uma função que determina a probabilidade dos eventos. O filtro utilizado no algoritmo serve como um mecanismo de controle da medida de inviabilidade, deixando de fora, da seleção dos melhores pontos, os piores pontos (isto é, os pontos proibidos pelo filtro corrente). Internamente, o algoritmo gera uma sequência (X_k) de iterandos (pontos de avaliação da função objetivo f e da medida de inviabilidade h) obtidos aleatoriamente de acordo com alguma distribuição de probabilidade.

Como propõe-se um algoritmo bastante genérico, no cálculo dos vetores aleatórios $X_k \in \mathcal{D}$ há uma certa liberdade na escolha da coleção de elementos aleatórios intermediários utilizada, denotada por $\{\Lambda_{k_j}\}_{1 \leq j \leq \ell_k}$, onde ℓ_k é um número positivo que depende da quantidade de elementos aleatórios que são gerados em cada iteração k . Cada elemento, Λ_{k_j} , da coleção pode ser uma variável aleatória ou vetor aleatório ou quaisquer outros tipos de elementos aleatórios definidos no espaço de probabilidade (Ω, \mathcal{F}, P) e cujas realizações seguem uma determinada distribuição de probabilidade. O oráculo associado à coleção de elementos aleatórios gerada até a iteração k é definido da mesma maneira como em (3.2). Além disso, $\sigma(\mathcal{O}_k)$ é a σ -álgebra gerada pelos elementos aleatórios do oráculo \mathcal{O}_k , e pode ser vista como uma representação de todas as informações que podem ser calculadas a partir dos elementos aleatórios de \mathcal{O}_k .

Vamos definir a região proibida de acordo com a regra de filtro inclinado, dada em (4.5), pois esta regra apresenta resultados mais fortes de convergência global que iremos utilizar na Seção 4.3.2.

A seguir, apresenta-se a proposta de um algoritmo geral estocástico de filtro.

Algoritmo 4.1 - *Algoritmo geral estocástico de filtro*

DADOS: $X_0 \in \mathcal{D}$, $X_0^* = X_0$, $\mathcal{O}_0 = \{X_0^*\}$, $\alpha \in (0, 1)$, $\varepsilon_h > 0$, $F_0 = \emptyset$, $\mathcal{F}_0 = \emptyset$

$k = 1$

ENQUANTO O CRITÉRIO DE PARADA NÃO FOR SATISFEITO

Defina $\bar{F}_{k-1} = F_{k-1} \cup \{(f_{k-1}^*, h_{k-1}^*)\}$ e $\bar{\mathcal{F}}_{k-1} = \mathcal{F}_{k-1} \cup \mathcal{R}_{k-1}$, com \mathcal{R}_{k-1} obtido de acordo com (4.5), onde $(f_j, h_j) = (f_{k-1}^*, h_{k-1}^*)$

Gere uma realização para cada elemento intermediário aleatório de acordo com alguma distribuição de probabilidade: $\{\Lambda_{k_j}\}_{1 \leq j \leq \ell_k}$

Defina $\mathcal{O}_k = \mathcal{O}_{k-1} \cup \{\Lambda_{k_j}\}_{1 \leq j \leq \ell_k}$

Calcule $X_k = \Theta(\mathcal{O}_k)$, de modo que $X_k \in \mathcal{D}$ e $X_k \notin \bar{\mathcal{F}}_{k-1}$

ATUALIZAÇÃO DO FILTRO:

SE $f(X_k) \leq f(X_{k-1}^*)$

$F_k = F_{k-1}$ e $\mathcal{F}_k = \mathcal{F}_{k-1}$ (iteração f)

SENÃO

$$F_k = \bar{F}_{k-1} \text{ e } \mathcal{F}_k = \bar{\mathcal{F}}_{k-1} \text{ (iteração } h\text{)}$$

ATUALIZAÇÃO DO MELHOR PONTO CORRENTE:

$$\text{SE } h(X_k) < h(X_{k-1}^*) + \varepsilon_h$$

$$X_k^* = X_k \text{ (iteração de sucesso)}$$

SENÃO

$$X_k^* = X_{k-1}^* \text{ (iteração de insucesso)}$$

$$k = k + 1$$

No início de cada iteração o par (f_{k-1}^*, h_{k-1}^*) é temporariamente introduzido no filtro corrente \bar{F}_{k-1} . Ao final da iteração, este par se tornará permanente no filtro apenas se a iteração não produzir uma redução em f , isto é, se a iteração for do tipo h . Se a iteração for do tipo f , então o novo elemento é descartado não havendo atualização do filtro. Além disso, podemos observar que se X_k , ou equivalentemente o par (f_k, h_k) , é viável, então qualquer ponto X não proibido pelo filtro deve satisfazer $f(X) < f(X_k)$, ou seja, a iteração será do tipo f e portanto este ponto não entrará no filtro, como ilustrado na Figura 4.5. É de realçar que este tipo de atualização é usual nos métodos de filtro [44]. O objetivo é tornar o filtro menos conservativo e permitir a exploração do espaço de procura próxima à fronteira da região viável.

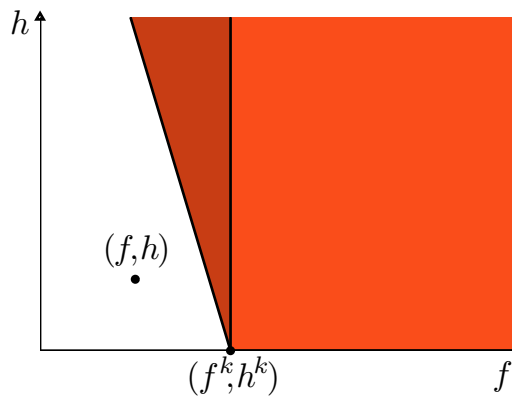


Figura 4.5: Caso em que X_k é viável

Na atualização do melhor ponto corrente, a prioridade é dada à redução da medida de

inviabilidade, com uma tolerância ε_h . É permitido que o valor da inviabilidade do iterando X_k possa aumentar um valor inferior a ε_h relativamente ao melhor ponto corrente. No entanto esses casos provocam uma redução no valor da função objetivo, uma vez que X_k é não proibido pelo filtro temporário corrente \bar{F}_{k-1} , que contém informação do melhor ponto corrente X_{k-1}^* . O objetivo desta estratégia, de permitir este pequeno aumento controlado na inviabilidade com redução na otimalidade, é promover a exploração de outras áreas potencialmente promissoras da região de procura e evitar que convirja para soluções ótimas locais. Note que, a partir do momento que a viabilidade é atingida, o melhor ponto será atualizado apenas se houver um decréscimo no valor da função objetivo. A Figura 4.6 ilustra a região do plano $f \times h$ que satisfaz o critério de seleção do melhor ponto corrente aqui proposto.

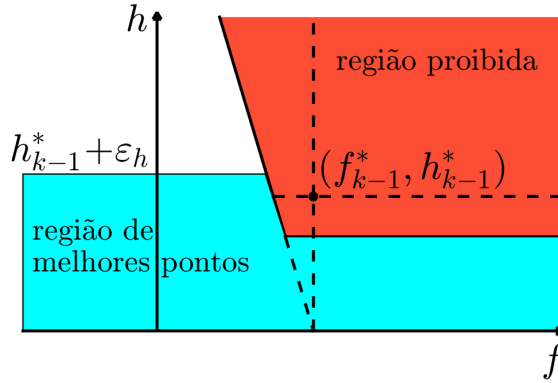


Figura 4.6: Critério de seleção do melhor ponto corrente

O Lema 4.2 estabelece que o Algoritmo 4.1 está bem definido no sentido de que sempre que um ponto corrente não satisfaz o critério de parada, um novo ponto não proibido pelo filtro corrente pode ser escolhido, a menos que o ponto corrente seja solução do problema.

Lema 4.2 *Considere o Algoritmo 4.1. Para todo $k \in \mathbb{N}$ tal que X_k não satisfaz o critério de parada, as seguintes afirmações são satisfeitas:*

- (i) *Temos $h_j > 0$, para todo $j \in \mathbb{N}$ tal que $(f_j, h_j) \in F_{k-1}$;*

(ii) Existe $X_k \notin \bar{\mathcal{F}}_{k-1}$.

Demonstração. Uma vez que a estrutura e atualização do filtro utilizadas no nosso algoritmo são as mesmas utilizadas em [44], a demonstração segue análoga a [44, Lema 8.4]. \square

Por outro lado, o próximo teorema estabelece que se for utilizada a regra de filtro inclinado, que é o caso do Algoritmo 4.1, então qualquer ponto de acumulação da sequência é viável, independente do procedimento interno utilizado para se obter os pontos de tal sequência.

Teorema 4.3 *Considere as sequências (X_k) e (X_k^*) geradas pelo Algoritmo 4.1. Então $h(X_k) \rightarrow 0$ e $h(X_k^*) \rightarrow 0$, ou seja, qualquer ponto de acumulação das sequências (X_k) e (X_k^*) é viável. Além disso, o número de iterações de insucesso é finito.*

Demonstração. Uma vez que a estrutura e atualização do filtro no nosso algoritmo são as mesmas utilizadas em [27] e que a sequência (X_k) é gerada de forma que $X_k \notin \bar{\mathcal{F}}_{k-1}$, então $h(X_k) \rightarrow 0$ segue de [27, Teorema 2.3].

Pelo mecanismo do algoritmo, X_k é selecionado como melhor ponto corrente se

$$h(X_k) < h(X_{k-1}^*) + \varepsilon_h, \quad (4.8)$$

onde $\varepsilon_h > 0$ é uma constante. Assim, de acordo com o algoritmo, se X_k satisfaz (4.8), então $h(X_k^*) = h(X_k)$. Por outro lado, se X_k não satisfaz (4.8), então $h(X_k^*) = h(X_{k-1}^*)$ e $h(X_k) \geq h(X_{k-1}^*) + \varepsilon_h$. Logo,

$$h(X_k^*) = h(X_{k-1}^*) \leq h(X_k) - \varepsilon_h < h(X_k).$$

Portanto, pelo Teorema do Confronto temos que $h(X_k^*) \rightarrow 0$.

Agora, suponha por contradição, que o número de iterações de insucesso, no Algoritmo 4.1, é infinito. Isto significa que existe uma subsequência de índices $(k_j)_{j \geq 1}$ satisfazendo,

$$h(X_{k_j}) \geq h(X_{k_j-1}^*) + \varepsilon_h \quad \text{e} \quad h(X_{k_j}^*) = h(X_{k_j-1}^*).$$

Logo, $h(X_{k_j}) \geq h(X_{k_j}^*) + \varepsilon_h$. Como provado anteriormente as sequências $h(X_k) \rightarrow 0$ e $h(X_k^*) \rightarrow 0$. Assim, fazendo $j \rightarrow \infty$, segue que $\varepsilon_h \leq 0$, o que contradiz o fato de ε_h ser uma constante estritamente positiva. \square

Uma vez estabelecido que o Algoritmo 4.1 está bem definido e que qualquer ponto de acumulação das sequências X_k e X_k^* geradas é viável, vamos discutir a convergência em probabilidade do algoritmo.

4.3.2 Convergência do algoritmo

Nesta seção, vamos analisar a convergência do Algoritmo 4.1, no sentido probabilístico, seguindo as ideias de [38]. Para isso, estamos assumindo que o algoritmo trabalha infinitamente, de modo a gerar as sequências (X_k) e (X_k^*) , sob as seguintes hipóteses.

H1 As sequências (X_k) e (X_k^*) permanecem no conjunto compacto \mathcal{D} .

H2 As funções f e h são contínuas em \mathcal{D} .

H3 Seja μ a medida de Lebesgue em \mathbb{R}^n . Para todo $\rho, \varepsilon > 0$, $\mu(T_{\rho,\varepsilon}) > 0$, onde

$$T_{\rho,\varepsilon} = \{x \in \mathcal{D} : h(x) \leq \rho \text{ e } f(x) < f^* + \varepsilon\}. \quad (4.9)$$

A Hipótese H1 é clássica e nos garante que a sequência (X_k^*) admite subsequência convergente para um ponto em \mathcal{D} . A Hipótese H2 nos garante que f é uniformemente contínua, uma vez que é contínua no compacto \mathcal{D} . Além disso, a imagem de \mathcal{D} por f é um conjunto compacto. A Hipótese H3 é bastante usual neste contexto. Tal hipótese é considerada na análise da convergência do algoritmo proposto em [46]. Em [37] é garantida a convergência quase certamente para um ponto no conjunto $S^* = \{x \in \mathcal{D}_V : f(x) \leq f_{egs}^*\}$, onde f_{egs}^* é o ínfimo sobre todos os valores f_t para os quais $\mu(\{z \in \mathcal{D} : h(z) \leq \rho, f(z) < f_t\}) > 0$, para todo $\rho > 0$.

A seguir apresentamos uma condição suficiente para a garantia da convergência do Algoritmo 4.1 em termos do ínfimo da função densidade condicional dos iterandos de vetores aleatórios. Para isso, considere o conjunto das iterações do tipo h dado por

$$\mathcal{K}_a = \{k \in \mathbb{N} : (f(X_{k-1}^*), h(X_{k-1}^*)) \text{ é adicionado ao filtro}\}. \quad (4.10)$$

Teorema 4.4 *Sejam μ a medida de Lebesgue em \mathbb{R}^n , x^* um minimizador global de f sobre \mathcal{D}_V , e suponha que as Hipóteses H1 - H3 sejam satisfeitas. Considere a sequência (X_k^*) , gerada pelo Algoritmo 4.1, e que para todo $\delta > 0$,*

$$\psi_{\mathcal{D}}(\delta) := \inf_{z \in \mathcal{D}} \mu(B(z, \delta) \cap \mathcal{D}) > 0.$$

Suponha que existe uma subsequência $(X_{k_j}^)$ de (X_k^*) tal que, para cada $j \geq 1$, $X_{k_j}^*$ tem densidade condicional $g_{k_j}(x \mid \sigma(\mathcal{O}_{(k_j)-1}))$ satisfazendo:*

$$\mu(\{x \in \mathcal{D} : G(x) = 0\}) = 0, \quad \text{onde} \quad G(x) := \inf_{j \geq 1} g_{k_j}(x \mid \sigma(\mathcal{O}_{(k_j)-1})). \quad (4.11)$$

Então, a sequência $(f(X_k^))$ tem um ponto de acumulação que é mínimo global de f sobre \mathcal{D}_V quase certamente. Além disso, se \mathcal{K}_a é finito, então qualquer ponto de acumulação da sequência $(f(X_k^*))$ é mínimo global de f sobre \mathcal{D} quase certamente, isto é, $f(X_k^*) \rightarrow f^*$ quase certamente. Neste último caso, se x^* é o único minimizador global, então $X_k^* \rightarrow x^*$ quase certamente.*

Demonstração. Considere sequências (ρ_ℓ) , (ε_ℓ) monótonas decrescentes, de termos positivos, convergentes para zero. Pela Hipótese H3 temos que $\mu(T_{\rho_\ell, \varepsilon_\ell}) > 0$, para todo $\ell \geq 1$. Assim, a probabilidade de que um ponto gerado aleatoriamente em \mathcal{D} esteja em $T_{\rho_\ell, \varepsilon_\ell}$ é estritamente positiva. Usando a hipótese e o item (i) do Lema 3.2 temos que existe uma subsequência $(X_{k_j}^*)$ tal que, para todo $j \geq 1$ e $z \in \mathcal{D}$, existe $v(z, \delta) \in (0, 1)$ satisfazendo,

$$P\left(X_{k_j}^* \in (B(z, \delta) \cap \mathcal{D}) \mid \sigma(\mathcal{O}_{(k_j)-1})\right) \geq v(z, \delta). \quad (4.12)$$

Em particular, (4.12) vale para \mathcal{D} restrito ao conjunto $T_{\rho_\ell, \varepsilon_\ell}$. Assim, para todo $\ell \geq 1$, existe k_{j_ℓ} tal que $X_{k_{j_\ell}}^* \in T_{\rho_\ell, \varepsilon_\ell}$ quase certamente. Definindo $S_\varepsilon := \{x \in \mathcal{D} : f(x) < f^* + \varepsilon\}$, temos que $T_{\rho_\ell, \varepsilon_\ell} \subset S_{\varepsilon_\ell}$ e, conseqüentemente, $X_{k_{j_\ell}}^* \in S_{\varepsilon_\ell}$ quase certamente.

Como a sequência (ε_ℓ) é decrescente e converge para zero, dado $\varepsilon > 0$, existe $\bar{\ell} \in \mathbb{N}$, tal que $\varepsilon > \varepsilon_{\bar{\ell}} > \dots > \varepsilon_\ell > \dots$, para todo $\ell \geq \bar{\ell}$, o que implica $S_\varepsilon \supseteq S_{\varepsilon_{\bar{\ell}}} \supseteq \dots \supseteq S_{\varepsilon_\ell} \supseteq \dots$. Portanto, $X_{k_{j_\ell}}^* \in S_\varepsilon$ para todo $\ell \geq \bar{\ell}$. Conseqüentemente,

$$X_{k_{j_\ell}}^* \notin S_\varepsilon \Rightarrow X_{k_{j_1}}^* \notin S_\varepsilon, X_{k_{j_2}}^* \notin S_\varepsilon, \dots, X_{k_{j_\ell}}^* \notin S_\varepsilon. \quad (4.13)$$

Pelo item (ii) do Lema 3.2, existe $L(\varepsilon) \in (0, 1)$, tal que, para todo $j \geq 1$, temos

$$P\left(X_{k_j}^* \in \mathcal{D} : f(X_{k_j}^*) < f(x^*) + \varepsilon \mid \sigma(\mathcal{O}_{(k_j)-1})\right) \geq L(\varepsilon). \quad (4.14)$$

Reescrevendo (4.14), em termos dos índices k_{j_ℓ} e do conjunto S_ε , temos

$$P\left(X_{k_{j_\ell}}^* \in S_\varepsilon \mid \sigma(\mathcal{O}_{(k_{j_\ell})-1})\right) \geq L(\varepsilon). \quad (4.15)$$

Por outro lado, de (2.4) e da Propriedade 2.4 (a), segue

$$\begin{aligned} P\left(X_{k_{j_1}}^* \notin S_\varepsilon, \dots, X_{k_{j_\ell}}^* \notin S_\varepsilon\right) &= \prod_{i=1}^{\ell} P\left(X_{k_{j_i}}^* \notin S_\varepsilon \mid X_{k_{j_1}}^* \notin S_\varepsilon, \dots, X_{k_{j_{(i-1)}}}^* \notin S_\varepsilon\right) \\ &= \prod_{i=1}^{\ell} \left(1 - P\left(X_{k_{j_i}}^* \in S_\varepsilon \mid X_{k_{j_1}}^* \notin S_\varepsilon, \dots, X_{k_{j_{(i-1)}}}^* \notin S_\varepsilon\right)\right). \end{aligned} \quad (4.16)$$

Do condicionamento dos elementos aleatórios em $\mathcal{O}_{k_{j_i}-1}$ e de (4.15), temos que

$$P\left(X_{k_{j_i}}^* \in S_\varepsilon \mid X_{k_{j_1}}^* \notin S_\varepsilon, \dots, X_{k_{j_{(i-1)}}}^* \notin S_\varepsilon\right) \geq L(\varepsilon). \quad (4.17)$$

Portanto, de (4.16) e (4.17) segue que

$$P\left(X_{k_{j_1}}^* \notin S_\varepsilon, \dots, X_{k_{j_\ell}}^* \notin S_\varepsilon\right) \leq (1 - L(\varepsilon))^\ell. \quad (4.18)$$

Logo, da Propriedade 2.4-(c), de (4.13) e (4.18), para cada $\ell \geq 1$, tem-se

$$\begin{aligned} 0 &\leq P\left(f\left(X_{k_{j_\ell}}^*\right) - f^* \geq \varepsilon\right) = P\left(f\left(X_{k_{j_\ell}}^*\right) \geq f^* + \varepsilon\right) = P\left(X_{k_{j_\ell}}^* \notin S_\varepsilon\right) \leq \\ &\leq P\left(X_{k_{j_1}}^* \notin S_\varepsilon, \dots, X_{k_{j_\ell}}^* \notin S_\varepsilon\right) \leq (1 - L(\varepsilon))^\ell. \end{aligned}$$

Portanto,

$$\lim_{\ell \rightarrow \infty} P\left(f\left(X_{k_{j_\ell}}^*\right) - f^* \geq \varepsilon\right) = 0.$$

Ou seja, $f\left(X_{k_{j_\ell}}^*\right) \rightarrow f^*$ em probabilidade. E pelo item (ii) do Teorema 2.40 segue que $f(X_{k_{j_\ell(i)}}^*) \rightarrow f^*$ quase certamente, com $i \rightarrow \infty$ para alguma subsequência de índices $(k_{j_\ell(i)})_{i \geq 1}$.

Agora, suponhamos que \mathcal{K}_a é finito. Logo, da maneira como o filtro é atualizado, temos que o número de iterações que provocam um acréscimo na função objetivo f é finito. Consequentemente, existe um índice \bar{k} tal que a sequência $(f(X_k^*))_{k \geq \bar{k}}$ é monótona não crescente, pois a partir de tal índice todas as iterações serão do tipo f . Além disso, dado que $f^* > -\infty$ temos que $\lim_{k \rightarrow \infty} f(X_k^*(\omega))$ existe para todo ponto amostral. Assim, da unicidade do limite podemos concluir que

$$f(X_k^*) \rightarrow f^* \quad \text{q.s.} \quad (4.19)$$

Resta agora mostrar que, se x^* existe e é único então $X_k^* \rightarrow x^*$ quase certamente. Para isso, fixado $\varepsilon > 0$ defina $\tilde{f} := \inf_{x \in \mathcal{D}_V, \|x - x^*\| \geq \varepsilon} f(x)$. Da unicidade de x^* , segue que $\tilde{f} > f^*$. De (4.19) temos que tal convergência ocorre a menos de um conjunto de medida nula; isto é, existe $\mathcal{N} \subset \Omega$ com $P(\mathcal{N}) = 0$ e tal que $f(X_k^*(\omega)) \rightarrow f(x^*)$, para todo $\omega \in \mathcal{N}^c$. Logo, da definição de limite de sequência, para todo $\omega \in \mathcal{N}^c$ existe um inteiro N tal que para todo $k \geq N$,

$$f(X_k^*(\omega)) - f(x^*) = |f(X_k^*(\omega)) - f(x^*)| < \tilde{f} - f(x^*),$$

ou seja,

$$f(X_k^*(\omega)) < \tilde{f}, \quad (4.20)$$

para todo $k \geq N$ e $\omega \in \mathcal{N}^c$.

Agora vamos mostrar que $\|X_k^*(\omega) - x^*\| < \varepsilon$ para todo $\omega \in \mathcal{N}^c$. De fato, suponha por contradição que existe $\omega \in \mathcal{N}^c$ tal que $\|X_k^*(\omega) - x^*\| \geq \varepsilon$. Da definição de \tilde{f} , segue que $f(X_k^*(\omega)) \geq \tilde{f}$, o que contradiz (4.20). Assim, temos que para todo $\omega \in \mathcal{N}^c$ existe um inteiro N tal que para todo $k \geq N$, $\|X_k^*(\omega) - x^*\| < \varepsilon$. Portanto, $X_k^*(\omega) \rightarrow x^*$ q.s. Consequentemente, X_k^* converge quase certamente para a solução do problema. \square

O próximo teorema, garante a convergência em probabilidade do Algoritmo 4.1 quando existe uma subsequência em que os iterandos são dados como a soma do melhor ponto obtido, até então, com um vetor aleatório cuja distribuição é multivariada normal.

Teorema 4.5 *Sejam μ a medida de Lebesgue em \mathbb{R}^n e x^* um minimizador global de f sobre \mathcal{D}_V . Suponha que as Hipóteses H1 - H3 sejam satisfeitas e que \mathcal{D} é um conjunto limitado*

satisfazendo $\psi_{\mathcal{D}}(\delta) := \inf_{z \in \mathcal{D}} \mu(B(z, \delta) \cap \mathcal{D}) > 0$ para todo $\delta > 0$. Considere a sequência (X_k^*) gerada pelo Algoritmo 4.1 e suponha que existe uma subsequência $(X_{k_j}^*)_{j \geq 1}$, tal que para cada $j \geq 1$, tem-se

$$X_{k_j}^* = X_{(k_j)-1}^* + W_j, \quad (4.21)$$

onde W_j é um vetor aleatório cuja distribuição condicional dado $\sigma(\mathcal{O}_{(k_j)-1})$ é multivariada normal com vetor de média nula e matriz de covariância V_j . Para cada $j \geq 1$, seja $\bar{\lambda}_j$ o menor autovalor de V_j . Se $\inf_{j \geq 1} \bar{\lambda}_j > 0$, então a sequência $f(X_k^*)$ tem um ponto de acumulação que é mínimo global de f sobre \mathcal{D}_V quase certamente. Além disso, se K_a é finito, então $f(X_k^*) \rightarrow f^*$ quase certamente. Neste último caso, se x^* é o único minimizador global, então $X_k^* \rightarrow x^*$ quase certamente.

Demonstração. Pelo mecanismo do algoritmo,

$$X_{(k_j)-1}^* = \sum_{i=1}^{(k_j)-1} X_i 1_E(X_i), \quad (4.22)$$

onde 1_E é a função indicadora e E é o evento definido por

$$E = [h(X_\ell) < h(X_t) + \varepsilon_h, \text{ para todo } t = 1, \dots, (k_j) - 1 \text{ e } \ell \text{ é o menor índice dos pontos não dominados pelo filtro } \bar{\mathcal{F}}_t \text{ com essa propriedade}].$$

A partir deste ponto de maneira análoga à demonstração do Teorema 3.6, podemos concluir que

$$\begin{aligned} G(x) &= \inf_{j \geq 1} g_{k_j}(x \mid \sigma(\mathcal{O}_{(k_j)-1})) \\ &\geq \gamma (\lambda_j^*)^{-n/2} \Psi \left(\frac{D^2}{\bar{\lambda}_j} \right) \\ &\geq \gamma \left(\sup_{\ell \geq 1} \lambda_\ell^* \right)^{-n/2} \Psi \left(\frac{D^2}{\inf_{\ell \geq 1} \bar{\lambda}_\ell} \right) > 0, \end{aligned}$$

onde $\gamma = (2\pi)^{-n/2}$, $\Psi = e^{-x/2}$, $D = \text{diam}(\mathcal{D})$ e λ_j^* é o maior autovalor de V_j . O que implica,

$$\mu(\{x \in \mathcal{D} : G(x) = 0\}) = 0.$$

E, assim, do Teorema 4.4 seguem os resultados. □

Portanto, qualquer algoritmo estocástico de filtro, nos moldes do Algoritmo 4.1, satisfazendo alguma das condições suficientes estudadas nesta seção converge quase certamente para uma solução do problema.

4.4 O Algoritmo de busca com escolha dinâmica das coordenadas e de filtro

O Algoritmo 4.1 é bastante geral no sentido de que a sua convergência quase certamente é garantida independentemente de qual distribuição de probabilidade tenha sido utilizada no cálculo dos iterandos. Nesta seção propomos o Algoritmo de Busca com Escolha Dinâmica das Coordenadas e de Filtro (FDDS), que se encaixa nos moldes do algoritmo geral, e tem como objetivo resolver problemas de otimização com restrições gerais do tipo (4.1) utilizando o método de filtro, descrito na Seção 4.2, para tratar a inviabilidade. O FDDS utiliza a estratégia de busca dinâmica das coordenadas do Algoritmo DDS, estudado no Capítulo 3, onde o iterando é selecionado a partir de um conjunto de pontos tentativos aleatórios obtidos perturbando todas, ou apenas algumas, das coordenadas do melhor ponto corrente.

Com o objetivo de uma exploração mais eficiente do espaço de busca, em cada iteração k o FDDS gera n_t múltiplos pontos aleatórios tentativos. Em seguida, é feita uma seleção para escolher o melhor entre os n_t 's pontos tentativos, X_k^{t*} , para substituir, ou não, o melhor ponto corrente, X_{k-1}^* , levando em consideração os valores das funções f e h nestes pontos. Antes de apresentar o algoritmo, vejamos alguns detalhes sobre alguns dos passos mais importantes do FDDS.

Conjunto de pontos iniciais

No Algoritmo FDDS é gerado um conjunto $I = \{X_1, \dots, X_{n_0}\} \subset \mathcal{D}$ de $n_0 = 2(n+1)$ pontos iniciais da mesma maneira que no Algoritmo DYCORS, de acordo com a Seção 3.3.3. Em seguida, seleciona-se o melhor ponto inicial X_0^* .

Definição 4.6 *O melhor ponto inicial X_0^* é o ponto do conjunto I que satisfaz uma das duas condições a seguir:*

- (i) *se todos os pontos de I são inviáveis, então o melhor ponto inicial será aquele com o menor valor de inviabilidade. Neste caso, se houver empate na inviabilidade, seleciona-se o ponto com o menor valor de função objetivo; ou*
- (ii) *se houver pontos viáveis em I , então o melhor ponto inicial será o ponto viável com menor valor de função objetivo.*

Coleção dos elementos aleatórios e função para cálculo dos iterandos

O processo de geração da coleção de elementos aleatórios utilizada no cálculo dos múltiplos pontos tentativos X_k^t , com $t = 1, \dots, n_t$, do Algoritmo FDDS, é feito de maneira análoga ao descrito na Seção 3.3.3. Isto é, $\{\Lambda_{kj}\}_{1 \leq j \leq 3n_t+1} = \{P_k, Z_k^t, \xi_k^t, \bar{i}^t\}$, onde P_k é a probabilidade de perturbação das coordenadas; Z_k^t é um vetor associado ao t -ésimo ponto tentativo, cujas componentes são números aleatórios distribuídos uniformemente no intervalo $[0, 1]$, e que juntamente com a probabilidade P_k serve para a tomada de decisão de perturbar ou não determinada componente do melhor ponto corrente; cada componente $(\xi_k^t)_i \sim N(0, (\lambda_{k-1})_i^2)$ é uma magnitude de perturbação adicionada à componente i do melhor ponto corrente, escolhida para ser perturbada, com uma distribuição normal de média 0 e desvio padrão $(\lambda_{k-1})_i$.

A probabilidade de perturbação das coordenadas do melhor ponto corrente, da iteração k , utilizada na coleção de elementos aleatórios é a mesma utilizada no Algoritmo DDS, e dada por

$$P_k = 1 - \frac{\ln(k)}{\ln(k_{max})}, \quad (4.23)$$

onde k_{max} é o número máximo de iterações.

A função determinística utilizada para calcular cada um dos n_t pontos tentativos, X_k^t , com $t = 1, \dots, n_t$, também é feita de maneira análoga ao descrito na Seção 3.3.3. Ou seja, numa iteração k , os n_t pontos tentativos são gerados perturbando-se apenas as coordenadas

do melhor ponto corrente, X_{k-1}^* , selecionadas por (3.38), através de (3.39) e (3.40). Iremos utilizar a notação Υ_k , para nos referir ao conjunto dos n_t 's pontos tentativos gerados em cada iteração k , isto é, $\Upsilon_k = \{X_k^1, X_k^2, \dots, X_k^{n_t}\}$.

Seleção do melhor ponto

O melhor ponto tentativo da iteração k , denotado por X_k^{t*} , é selecionado como segue.

Definição 4.7 (Melhor ponto tentativo) *Fixada uma iteração k , o melhor dos pontos tentativos do conjunto Υ_k , denotado por X_k^{t*} , é um ponto não dominado pelo filtro temporário corrente $\bar{\mathcal{F}}_{k-1}$ e que satisfaz uma das duas condições a seguir:*

- (i) *se todos os múltiplos pontos tentativos são inviáveis, então o melhor ponto tentativo será aquele com o menor valor de inviabilidade. Neste caso, se houver empate na inviabilidade, seleciona-se o ponto com o menor valor de função objetivo; ou*
- (ii) *se houver pontos tentativos viáveis, então o melhor ponto tentativo será o ponto viável com menor valor de função objetivo.*

A Figura 4.7 ilustra as duas situações da definição do melhor ponto tentativo para um conjunto de três pontos tentativos, dado por $\Upsilon_k = \{X_k^1, X_k^2, X_k^3\}$. Para não pesar na notação, estamos representando o par $(f(X_k^t), h(X_k^t))$ por apenas (X_k^t) .

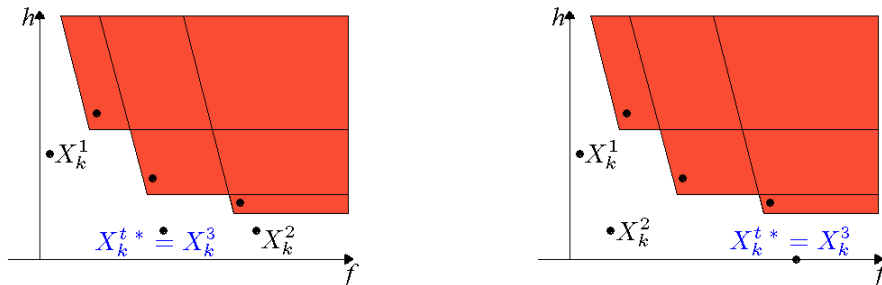


Figura 4.7: Seleção do melhor ponto tentativo no FDDS

Em seguida o melhor ponto tentativo não dominado pelo filtro corrente, X_k^{t*} , é comparado com o melhor ponto corrente, X_{k-1}^* . Se o melhor ponto tentativo na iteração k não for considerado melhor do que o melhor ponto corrente, de acordo com a seleção de melhor ponto do Algoritmo 4.1, então X_{k-1}^* é preservado para a próxima iteração.

4.4.1 Algoritmo FDDS

Nesta seção vamos descrever o Algoritmo FDDS na sua forma geral.

Algoritmo 4.2 - Algoritmo FDDS

DADOS: $\alpha \in (0, 1)$; $\eta \in (0, 1)$; $\tau_{suc}, \tau_{falha}, n_t, k_{max} \in \mathbb{N}$; $\lambda_0, \varepsilon_f, \varepsilon_h > 0$;

$$c_{suc} = c_{falha} = 0; F_0 = \mathcal{F}_0 = \emptyset; n_0 = 2(n + 1)$$

Pontos iniciais: $I = \{X_1, \dots, X_{n_0}\} \subset \mathcal{D}$, gerados de acordo com (3.35)

Selecione o melhor ponto, X_0^* , de I de acordo com a Definição 4.6 e defina $\mathcal{O}_0 = \{X_0^*\}$

$k = 1$ e $n_f = n_0$

ENQUANTO O CRITÉRIO DE PARADA NÃO FOR SATISFEITO

Defina $\bar{F}_{k-1} = F_{k-1} \cup \{(f_{k-1}^*, h_{k-1}^*)\}$ e $\bar{\mathcal{F}}_{k-1} = \mathcal{F}_{k-1} \cup \mathcal{R}_{k-1}$, com \mathcal{R}_{k-1} obtido de acordo com (4.5), onde $(f_j, h_j) = (f_{k-1}^*, h_{k-1}^*)$

Gere $\{\Lambda_{k_j}\}_{1 \leq j \leq 3n_t+1} = \{P_k, Z_k^t, \xi_k^t, i^t\}$, de acordo com (3.36)

Defina $\mathcal{O}_k = \mathcal{O}_{k-1} \cup \{\Lambda_{k_j}\}_{1 \leq j \leq 3n_t+1}$

Calcule o conjunto J_k^t , com $t = 1, \dots, n_t$, de acordo com (3.38)

Calcule $\Upsilon_k = \{X_k^t, \text{ com } t = 1, \dots, n_t\}$, onde $X_k^t = \Theta^t(\mathcal{O}_k) \in \mathcal{D}$, de acordo com (3.40)

Selecione o subconjunto $\Upsilon_k^* = \{X_k^t \in \Upsilon_k : X_k^t \notin \bar{\mathcal{F}}_{k-1}\}$

SE $\Upsilon_k^* \neq \emptyset$

Selecione $X_k^{t*} \notin \Upsilon_k^*$ de acordo com a Definição 4.7 e faça $n_f = n_f + n_t$

ATUALIZAÇÃO DO FILTRO:

SE $f(X_k^{t*}) \leq f(X_{k-1}^*)$

$F_k = F_{k-1}$ e $\mathcal{F}_k = \mathcal{F}_{k-1}$ (iteração f)

SENÃO

$F_k = \bar{F}_{k-1}$ e $\mathcal{F}_k = \bar{\mathcal{F}}_{k-1}$ (iteração h)

ATUALIZAÇÃO DO MELHOR PONTO CORRENTE:

$$\text{SE } h(X_k) < h(X_{k-1}^*) + \varepsilon_h$$

$$X_k^* = X_k^{t*} \quad (\text{iteração de sucesso})$$

$$\text{SE } (f(X_{k-1}^*) - f(X_k^{t*})) > \eta |f(X_{k-1}^*)|$$

$$c_{suc} = c_{suc} + 1 \text{ e } c_{falha} = 0$$

SENÃO

$$X_k^* = X_{k-1}^* \quad (\text{iteração de insucesso})$$

$$c_{falha} = c_{falha} + 1 \text{ e } c_{suc} = 0$$

SENÃO

$$X_k^* = X_{k-1}^*$$

AJUSTE DO TAMANHO DO PASSO:

$$\text{SE } c_{falha} \geq \tau_{falha}$$

$$\lambda_k = \max(\lambda_{k-1}/2, \lambda_0), c_r = c_r + 1 \text{ e } c_{falha} = 0$$

$$\text{SE } c_{suc} \geq \tau_{suc}$$

$$\lambda_k = \min(2\lambda_{k-1}, \lambda_0) \text{ e } c_{suc} = 0$$

$$k = k + 1$$

É de salientar que, no Algoritmo FDDS é feito um ajuste do tamanho do passo através do controle do desvio padrão, λ_k , associado ao tamanho da magnitude de perturbação do passo. Na prática, considera-se o parâmetro inicializado por $\lambda_0 = 0.2\Delta_{min}$, onde $\Delta_{min} = \min_{1 \leq i \leq n} (u_i - \ell_i)$. Este parâmetro exerce o controle sobre a aleatoriedade ou, de forma equivalente, sobre a diversidade dos pontos gerados. No início do processo iterativo, este parâmetro deve tomar valores grandes, de modo a forçar o algoritmo a aumentar a diversidade dos pontos. Ao longo do processo iterativo seu valor é ajustado conforme a iteração seja considerada de sucesso ou insucesso, uma vez que o esforço se concentra na exploração. Para auxiliar no processo de ajuste do tamanho do passo, usa-se os contadores de sucesso c_{suc} e de falha c_{falha} . No entanto, um sucesso é contabilizado se adicionalmente o melhor ponto tentativo provocar um decréscimo suficiente na função objetivo, ou seja, se $(f(X_{k-1}^*) - f(X_k^{t*})) > \eta |f(X_{k-1}^*)|$. Além disso, também são utilizados, o parâmetro de

tolerância máxima de falhas ou insucessos, τ_{falha} , para decidir se o tamanho do passo é reduzido, e a tolerância máxima de sucessos, τ_{suc} , para decidir se o tamanho do passo é aumentado.

4.4.2 Convergência em probabilidade

Nesta seção mostramos que o Algoritmo FDDS satisfaz as condições suficientes estabelecidas no Teorema 4.5, que juntamente com o Teorema 4.3, garantem a sua convergência quase certamente para uma solução.

Como já referido, o Algoritmo FDDS gera seus iterandos perturbando-se algumas, ou todas, as componentes do melhor ponto corrente. A escolha das coordenadas a serem perturbadas segue uma distribuição uniforme, enquanto que a magnitude de perturbação segue uma distribuição normal. Primeiro, vamos provar que a sequência gerada pelo Algoritmo FDDS admite uma subsequência onde todas as componentes do vetor iterando possuem perturbação aleatória com distribuição normal. Veremos, no lema a seguir, que uma hipótese suficiente para a existência de tal subsequência é que a probabilidade associada em cada iteração $P_k \in (0, 1]$ seja maior que uma constante $\bar{p} > 0$.

Uma vez que os iterandos no FDDS são gerados da mesma maneira que no Algoritmo DDS, a demonstração do próximo resultado segue de maneira análoga ao Lema 3.8, apresentado no Capítulo 3 na prova de convergência do Algoritmo DDS.

Lema 4.8 *Consideremos a sequência (X_k^*) gerada pelo Algoritmo FDDS com $P_k \in (0, 1]$ e $P_k \geq \bar{p}$, onde \bar{p} é uma constante estritamente positiva. Existe uma subsequência $(X_{k_j}^*)$ em \mathcal{D} onde todas as componentes de todos os seus termos possuem perturbação aleatória com distribuição normal.*

Demonstração. Segue de maneira análoga à demonstração do Lema 3.8, considerando o evento de sucesso A_k como sendo o evento onde todas as coordenadas do termo X_k^* possuem perturbação aleatória com distribuição normal. □

Isto significa que, de acordo com a construção da sequência (X_k^*) , no Algoritmo FDDS, existe uma subsequência $(X_{k_j}^*)$ satisfazendo (4.21).

Teorema 4.9 *Sejam μ a medida de Lebesgue em \mathbb{R}^n , \mathcal{D} um conjunto limitado tal que $\psi_{\mathcal{D}}(\delta) > 0$, para todo $\delta > 0$, x^* um minimizador global de f sobre \mathcal{D}_V , \bar{p} uma constante estritamente positiva, e suponha que as Hipóteses H1 - H3 sejam satisfeitas. Se (X_k^*) é uma sequência gerada pelo Algoritmo FDDS com $P_k \in (0, 1]$ e $P_k \geq \bar{p}$, então a sequência $f(X_k^*)$ tem um ponto de acumulação que é mínimo global de f sobre \mathcal{D}_V quase certamente. Além disso, se \mathcal{K}_a é finito, então $f(X_k^*) \rightarrow f^*$ quase certamente. Neste último caso, se x^* é o único minimizador global, então $X_k^* \rightarrow x^*$ quase certamente.*

Demonstração. Pelo Lema 4.8 existe uma subsequência $(X_{k_j}^*)$ onde todas as coordenadas de cada termo de tal subsequência possuem perturbação aleatória com distribuição normal. Assim, da construção da sequência no Algoritmo FDDS, temos que $X_{k_j}^*$ é dada por (3.32), onde cada componente $(\xi_{k_j})_i \sim N(0, (\lambda_{k_j})_i^2)$ segue uma distribuição normal de média zero e desvio padrão dado pela raiz quadrada dos autovalores da matriz de covariância. Deste modo, $\xi_{k_j} \sim N(0, V_j)$ é um vetor aleatório cuja distribuição condicional, dado $\sigma(\mathcal{O}_{(k_j)-1})$, é a distribuição normal multivariada com vetor de média nula e matriz de covariância V_j dada por

$$V_j = \text{Cov}(\xi_{k_j}) = \text{diag}((\lambda_{k_j})_1^2, \dots, (\lambda_{k_j})_n^2).$$

Neste caso, temos que os autovalores de V_j são as variâncias $(\lambda_{k_j})_1^2, \dots, (\lambda_{k_j})_n^2$ das perturbações aleatórias normais adicionadas à cada componente de $X_{(k_j)-1}^*$. Assim, o menor autovalor de V_j é dado por

$$\bar{\lambda}_j := \min_{1 \leq i \leq n} (\lambda_{k_j})_i^2 > 0.$$

Portanto, o resultado segue do Teorema 4.5.

□

4.5 Algoritmo FDDS usando modelos de aproximação por funções de base radial

Nesta seção propõe-se o Algoritmo FDDSRBF, o qual é um algoritmo estocástico que usa modelos de aproximação por funções de base radial para resolver problemas com restrições

gerais e que utiliza o método de filtro para lidar com a inviabilidade. É de relembrar que, devido à estratégia de calcular múltiplos pontos tentativos, em cada iteração k , o FDDS gasta um número elevado de avaliações da função objetivo para resolver um determinado problema. Com o intuito de diminuir o elevado número de avaliações de f , e assim obter um algoritmo de baixo custo computacional sem abdicar da estratégia dos múltiplos pontos tentativos, recorreu-se a uma aproximação da função objetivo f através de interpolação por funções de base radial, e que foi inspirado no estudo do Algoritmo DYCORS apresentado no Capítulo 3.

Assim, incorporamos o modelo de interpolação por funções de base radial (RBF) ao FDDS, dando origem ao Algoritmo de Busca com Escolha Dinâmica das Coordenadas e de Filtro usando interpolação por funções de base radial (FDDSRBF). A aproximação por funções de base radial será utilizada para substituir a função objetivo na seleção do melhor entre os múltiplos pontos tentativos gerados em cada iteração, X_k^{t*} . Para isso, utilizamos o modelo RBF cúbico S , para interpolação da função objetivo, apresentado na Seção 3.3.3. No Algoritmo FDDSRBF, inicialmente são gerados $n_0 = 2(n+1)$ pontos iniciais da mesma forma que no Algoritmo FDDS. Em seguida, seleciona-se o melhor ponto inicial X_0^* de acordo com a Definição 4.6.

Coleção dos elementos aleatórios e função para cálculo dos iterandos

Assim como no FDDS, no Algoritmo FDDSRBF o processo de geração da coleção de elementos aleatórios, $\{\Lambda_{k_j}\}_{1 \leq j \leq 3n_t+1}$, utilizada no cálculo dos múltiplos pontos tentativos X_k^t , com $t = 1, \dots, n_t$, é dada por (3.36). Porém, neste caso a probabilidade é calculada em função do número de avaliações da função objetivo de acordo com (3.37), como feito no Algoritmo DYCORS, Seção 3.3.3.

A função determinística utilizada para calcular cada um dos n_t pontos tentativos, X_k^t , com $t = 1, \dots, n_t$, também é feita de maneira análoga ao Algoritmo FDDS, através de (3.39) e (3.40).

Seleção do melhor ponto tentativo usando RBF

Como referido, na seleção do k -ésimo melhor ponto tentativo X_k^{t*} do Algoritmo FDDSRBF, é utilizado um modelo de interpolação por funções de base radial descrito em [42] para prever os valores da função objetivo avaliados em tais pontos tentativos, no entanto outras aproximações da função objetivo podem ser utilizadas. Para isso, utilizamos o modelo RBF cúbico dado por (3.41) e cujos coeficientes são obtidos através da solução do sistema (3.42), como apresentado na Seção 3.3.3. Esta estratégia do uso do modelo RBF possibilita uma economia no número de avaliações da função objetivo mesmo que trabalhando com múltiplos pontos tentativos em cada iteração. Como o problema envolve restrições gerais, é também usada a função medida de inviabilidade h , dada por (4.2), na seleção de X_k^{t*} . Ou seja, a seleção do melhor ponto tentativo no FDDSRBF é feita levando em consideração os valores aproximados da função objetivo S e os valores da função de medida de inviabilidade h .

Assim, para selecionar o melhor ponto tentativo, da iteração k , propomos um filtro auxiliar, denotado por F_S , baseado na medida S (em vez de f) e na medida de inviabilidade h . Para isso, o filtro F_S é definido como sendo um conjunto de pares $(S(X_k^t), h(X_k^t))$ não dominados entre si, com $X_k^t \in \Upsilon_k$, correspondentes à coleção de pontos tentativos aleatórios gerada em cada iteração. Para simplificar a notação, vamos denotar o par $(S(X_k^t), h(X_k^t))$ por (S^t, h^t) . Considera-se o conceito de dominância apresentado na Seção 4.2, trocando-se os valores $f(X_k^t)$ por $S(X_k^t)$ e, em cada iteração k , o filtro auxiliar F_S é inicializado vazio.

Para fixar as ideias, um ponto Y , ou equivalentemente o par $(S(Y), h(Y))$, é não dominado pelo filtro auxiliar F_S se para todo $(S_k^t, h^t) \in F_S$, temos

$$S_k(Y) + \alpha h(Y) < S_k^t \quad \text{ou} \quad h(Y) < (1 - \alpha)h^t. \quad (4.24)$$

Neste contexto, dado que em cada iteração o filtro é inicializado vazio, não existe o denominado filtro temporário. Relativamente à sua atualização, será adicionado ao filtro auxiliar F_S cada ponto tentativo, gerado na iteração k , não dominado por F_S , independente desse ponto ter produzido um acréscimo ou não em S . Neste caso, um ponto tentativo viável

pode ser incluído no filtro F_S . Portanto, no Algoritmo FDDSRBF o melhor ponto tentativo X_k^{t*} da iteração k é selecionado de acordo com a seguinte definição.

Definição 4.10 *O melhor ponto tentativo, da iteração k , é o ponto tentativo com o menor valor de medida de inviabilidade, não dominado pelo filtro auxiliar F_S , de acordo com (4.24).*

A Figura 4.8 ilustra uma situação de seleção do melhor ponto tentativo no Algoritmo FDDSRBF para um conjunto de quatro pontos tentativos, dado por $\Upsilon_k = \{X_k^1, X_k^2, X_k^3, X_k^4\}$. Para não pesar na notação, estamos representando o par $(S(X_k^t), h(X_k^t))$ por apenas X_k^t .

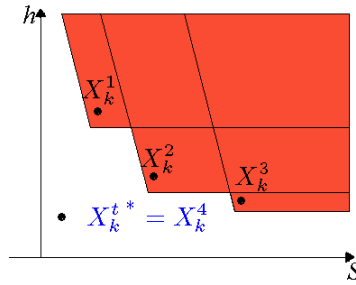


Figura 4.8: Seleção do melhor ponto tentativo no FDDSRBF

É de salientar que outro método para seleção do melhor ponto tentativo pode ser utilizado, porém a ideia principal neste procedimento de seleção é não desperdiçar gasto de avaliações da função objetivo com pontos tentativos não promissores.

Após a seleção do melhor ponto tentativo X_k^{t*} , o Algoritmo FDDSRBF finalmente verifica se tal ponto é não proibido pelo filtro temporário corrente. Para isso, calculamos $f(X_k^{t*})$, contabilizando esta avaliação de f . Em caso afirmativo, isto é se $X_k^{t*} \notin \bar{\mathcal{F}}_{k-1}$, adicionamos X_k^{t*} ao conjunto \mathcal{A}_k , dos pontos utilizados pelo modelo RBF para aproximar a função objetivo na seleção do melhor ponto tentativo da iteração seguinte. Em seguida, verificamos se X_k^{t*} satisfaz o critério de atualização do melhor ponto corrente para atualizar tal ponto. Caso contrário, permanece o melhor ponto corrente da iteração anterior.

4.5.1 Algoritmo FDDSRBF

A seguir apresentamos a configuração geral do Algoritmo FDDSRBF.

Algoritmo 4.3 - *Algoritmo FDDSRBF*

DADOS: $\alpha \in (0, 1)$; $\eta \in (0, 1)$; $\tau_{suc}, \tau_{falha}, n_t, N_{max} \in \mathbb{N}$; $\lambda_0, \varepsilon_f, \varepsilon_h > 0$; $\eta = 10^{-3}$

$$c_r = c_{suc} = c_{falha} = 0, F_0 = \emptyset, \mathcal{F}_0 = \emptyset, S_0 = \emptyset; n_0 = 2(n + 1)$$

Gere o conjunto de pontos iniciais I de acordo com (3.35)

Calcule $f(X_j)$ para cada $j = 1, \dots, n_0$, selecione X_0^* por 4.6 e faça $\mathcal{O}_0 = \{X_0^*\}$

$k = 1, n_f = n_0$ e $\mathcal{A}_k = I$

ENQUANTO O CRITÉRIO DE PARADA NÃO FOR SATISFEITO

Defina $\bar{F}_{k-1} = F_{k-1} \cup \{(f_{k-1}^*, h_{k-1}^*)\}$ e

$\bar{\mathcal{F}}_{k-1} = \mathcal{F}_{k-1} \cup \mathcal{R}_{k-1}$, com \mathcal{R}_{k-1} obtido de acordo com (4.5)

Ajuste ou atualize o modelo RBF (S) usando os pontos do conjunto \mathcal{A}_k

de acordo com (3.41) e (3.42)

Gere $\{\Lambda_{k_j}\}_{1 \leq j \leq 3n_t+1} = \{P_k, Z_k^t, \xi_k^t, \bar{t}^t\}$, com $t = 1, \dots, n_t$, de acordo com (3.36)

Defina $\mathcal{O}_k := \mathcal{O}_{k-1} \cup \{\Lambda_{k_j}\}_{1 \leq j \leq 3n_t+1}$

Calcule o conjunto J_k^t , com $t = 1, \dots, n_t$, de acordo com (3.38)

Calcule, $\Upsilon_k = \{X_k^t, t = 1, \dots, n_t\}$, onde $X_k^t = \Theta^t(\mathcal{O}_k) \in \mathcal{D}$, de acordo com (3.40)

Para cada $Y \in \Upsilon_k$ calcule $S(Y)$ e $h(Y)$

Selecione X_k^{t*} de acordo com a Definição 4.10, calcule $f(X_k^{t*})$ e $n_f = n_f + 1$

SE $X_k^{t*} \notin \bar{\mathcal{F}}_{k-1}$

$$\mathcal{A}_{k+1} = \mathcal{A}_k \cup \{X_k^{t*}\}$$

ATUALIZAÇÃO DO FILTRO:

$$\text{SE } f(X_k^{t*}) \leq f(X_{k-1}^*)$$

$$F_k = F_{k-1} \text{ e } \mathcal{F}_k = \mathcal{F}_{k-1}$$

SENÃO

$$F_k = \bar{F}_{k-1} \text{ e } \mathcal{F}_k = \bar{\mathcal{F}}_{k-1}$$

ATUALIZAÇÃO DO MELHOR PONTO CORRENTE:

$$\text{SE } h(X_k) < h(X_{k-1}^*) + \varepsilon_h$$

$$X_k^* = X_k^{t^*}$$

$$\text{SE } (f(X_{k-1}^*) - f(X_k^{t^*})) > \eta |f(X_{k-1}^*)|$$

$$c_{suc} = c_{suc} + 1 \text{ e } c_{falha} = 0$$

SENÃO

$$X_k^* = X_{k-1}^*, c_{falha} = c_{falha} + 1 \text{ e } c_{suc} = 0$$

SENÃO

$$X_k^* = X_{k-1}^*$$

AJUSTE DO TAMANHO DO PASSO:

$$\text{SE } c_{falha} \geq \tau_{falha}$$

$$\lambda_k = \max(\lambda_{k-1}/2, \lambda_0), c_r = c_r + 1 \text{ e } c_{falha} = 0$$

$$\text{SE } c_{suc} \geq \tau_{suc}$$

$$\lambda_k = \min(2\lambda_{k-1}, \lambda_0) \text{ e } c_{suc} = 0$$

$$k = k + 1.$$

4.5.2 Convergência em probabilidade

Uma vez que os pontos tentativos no Algoritmo FDDSRBF são gerados da mesma maneira que no Algoritmo FDDS, diferindo apenas na maneira como o melhor ponto tentativo é selecionado, segue que a sequência (X_k^*) gerada pelo FDDSRBF satisfaz o Lema 4.8 e, portanto, admite uma subsequência cujos termos possuem todas as componentes com perturbação aleatória seguindo uma distribuição normal. Desta forma, a demonstração da convergência em probabilidade do FDDSRBF segue de forma análoga à convergência em probabilidade do FDDS, apresentada na Seção 4.4.2. Ou seja, a sequência (X_k^*) gerada pelo Algoritmo FDDSRBF também satisfaz o Teorema 4.9, donde segue a convergência em probabilidade do algoritmo.

Capítulo 5

Experimentos numéricos

Neste capítulo reportamos experimentos numéricos para ilustrar o desempenho prático dos Algoritmos **FDDS** e **FDDSRBF**, propostos nas Seções 4.4 e 4.5. Os resultados obtidos ao resolver três conjuntos de problemas teste descritos em [3, 20, 47], serão comparados com os resultados dos seguintes algoritmos existentes na literatura, a saber os Algoritmos **ALtype 3**, **A2-SF**, **EPGO** e **P-BS AFS**, apresentados, respectivamente, em [18, 30, 10, 46].

Designamos por **ALtype 3** o **Algoritmo 2** (com a função de penalidade denotada por 3) apresentado em [18], o qual é um algoritmo de otimização global para problemas de PNL, que consiste num método de penalidade baseado em funções Lagrangianas aumentadas e que recorre a um método de iniciação múltipla que incorpora como procedimento local o método **GENCAN** [5] para resolver os subproblemas. Em [18], o Algoritmo **ALtype 3** foi testado usando 7 dos problemas descritos em [3].

Em [30], é proposto o **A2-SF**, um algoritmo estocástico que utiliza a ideia de escolha dinâmica das coordenadas, a perturbar, do **DDS** e o método de filtro para tratar a inviabilidade. Com o objetivo de explorar outras áreas promissoras de região de busca, o **A2-SF** incorpora uma estratégia chamada “Poll-Search”. Caso a iteração falhe na busca pelo melhor ponto, antes de declarar a iteração como insucesso invoca-se este procedimento, que consiste em procurar novos pontos possivelmente promissores na vizinhança do ponto menos inviável encontrado até a iteração corrente.

O P-BS AFS é um método estocástico de otimização global baseado numa população de pontos que simula o comportamento de um cardume na água. Este método usa a metodologia de filtro para aceitar, em cada iteração, uma população de pontos tentativos não dominados da população sempre que melhorem a violação das restrições ou a função objetivo em relação às soluções atuais.

O EPGO, é um algoritmo de otimização global que recorre a uma abordagem de penalidade exata. Neste método, é usado o algoritmo determinístico DIRECT (ver [29]) para resolver globalmente os subproblemas com restrições de limites simples. A principal característica do EPGO é que ele incorpora uma regra de atualização para o parâmetro de penalidade que, sob certas hipóteses, ocorre apenas um número finito de vezes.

5.1 Detalhes de implementação

Para os experimentos numéricos realizados com os algoritmos propostos, o critério de parada adotado é baseado nos valores da medida de inviabilidade e da função objetivo. Assim, os algoritmos param quando o melhor ponto X_k^* , da iteração k , satisfaz

$$f(X_k^*) \leq f^* + \varepsilon_f \quad \text{e} \quad h(X_k^*) \leq \varepsilon_h, \quad (5.1)$$

onde ε_f é a precisão do erro exigida no valor da função objetivo, ε_h é a tolerância da medida de inviabilidade e f^* é o valor ótimo conhecido. Para problemas em que f^* não é conhecido, a diferença absoluta entre os valores da função objetivo de iterações consecutivas são comparadas com ε_f para decidir parar o algoritmo. Além disso, se (5.1) não for satisfeita, o FDDS e FDDSRBF têm uma alternativa de parada baseada no número máximo de iterações $k_{max} = 1000$. As precisões exigidas para as funções objetivo e medida de inviabilidade foram as mesmas utilizadas em [46], $\varepsilon_f = 10^{-4}$ e $\varepsilon_h = 10^{-8}$, respectivamente.

Os resultados numéricos dos Algoritmos A2-SF e P-BF AFS foram retirados de [30, 46], respectivamente, para o mesmo critério de parada e número de rodadas utilizados nos experimentos numéricos com os Algoritmos FDDS e FDDSRBF. Os resultados numéricos do Algoritmo ALtype 3 foram retirados de [18] para uma precisão da função objetivo e

tolerância de inviabilidade, ambos, de 10^{-4} . Os resultados numéricos do Algoritmo EPGO foram retirados de [10].

Seguindo as ideias de [46], os problemas com restrições de igualdade $c_i(x) = 0$, para $i \in \mathcal{E}$, foram reformulados como desigualdades, usando $c_i(x) - \delta \leq 0$ e $-c_i(x) - \delta \leq 0$, onde $\delta = 10^{-4}$.

O número de pontos tentativos gerados, em cada iteração, varia de acordo com o número de variáveis de decisão (n) do problema. Os valores utilizados nos experimentos numéricos foram $n_t = 5n$, tal como usado em [30], para o Algoritmo FDDS, e $n_t = \min\{100n, 5000\}$, como sugerido por [42], para o Algoritmo FDDSRBF. Os parâmetros tolerância máxima de falhas ou insucessos, $\tau_{falha} = \max\{5, n\}$, e tolerância máxima de sucessos, $\tau_{suc} = 3$, foram fixados como sugerido em [42].

5.2 Resultados numéricos

Para validar e avaliar o desempenho numérico dos Algoritmos FDDS e FDDSRBF, vamos usar três conjuntos de problemas teste de otimização global com restrições gerais. O primeiro conjunto envolve 20 problemas de otimização global com restrições gerais, identificados por 1,2a, 2b, 2c, 2d, 3a, 3b, 4 . . . , 16, descritos em [3], em que número de variáveis de decisão variam entre 2 e 10. O segundo, é formado por um conjunto de 13 problemas teste com o número de variáveis de decisão variando entre 2 e 20, identificados por G1, G2, G3, . . . , G13 e descritos em [20]. O último conjunto, compreende 9 problemas conhecidos como problemas de engenharia, apresentados em [47], e o número de variáveis de decisão variam entre 2 e 8. Os testes foram realizados num notebook ASUSTek Intel Core i7-6700HQ, CPU 2.60GHz, com 16GB RAM, 64-bit, usando Matlab R2015a.

Uma vez que os algoritmos a serem analisados são estocásticos, para haver significância estatística dos resultados os mesmos devem ser rodados um número significativo de vezes. Neste caso, os resultados dos algoritmos foram obtidos após 30 rodadas independentes. Para medir a tendência central dos resultados obtidos pelos algoritmos, sobre as 30 rodadas, utilizamos a mediana ou a média tomadas sobre todas as rodadas.

Para avaliar e comparar o desempenho dos algoritmos vamos utilizar o perfil de desempenho, inicialmente proposto em [11]. Os perfis de desempenho permitem representar graficamente uma comparação, em termos de eficiência e robustez, entre os vários algoritmos para um conjunto de problemas teste. O gráfico do perfil de desempenho representa a função de distribuição da razão de desempenho com base numa dada métrica. Dolan e Moré [11] propuseram como métrica o tempo de computação necessário para resolver um problema, mas outras métricas poderão ser usadas. No caso de otimização sem derivadas, salienta-se que a tarefa mais dispendiosa para qualquer algoritmo é o cálculo do valor da função objetivo, pelo que o número de avaliações da função é a métrica mais usada para avaliar o desempenho dos algoritmos. Em seguida, apresentamos uma breve discussão sobre a implementação desta avaliação de desempenho.

Considere-se \mathcal{P} o conjunto dos n_p problemas teste utilizados e \mathcal{S} o conjunto de n_s algoritmos usados no estudo comparativo. Para cada problema $p \in \mathcal{P}$ e algoritmo $s \in \mathcal{S}$, seja $m_{p,s}$ a métrica necessária para a resolução de p utilizando-se s . Adotamos o número médio de avaliações da função objetivo, $n_{f_{medio}}$, como métrica de desempenho na comparação dos experimentos. Porém, para que a comparação entre os algoritmos fosse justa, também levamos em consideração, para o cálculo da métrica $m_{p,s}$, a capacidade de determinado algoritmo resolver um problema, como sugerido em [4]. Por exemplo, um algoritmo pode ter obtido o menor valor da métrica (menor número médio de avaliações da função objetivo), mas não ter conseguido obter uma solução viável. Assim, consideramos que um algoritmo resolve um determinado problema p se ele encontra um ponto viável e se satisfaz a condição

$$\frac{f - f_{\min}}{\max\{1, |f_{\min}|\}} \leq 10^{-3}, \quad (5.2)$$

onde f pode ser o melhor valor da função objetivo obtido pelo algoritmo nas 30 rodadas (f_{otm}) e f_{\min} o menor valor da função objetivo encontrada entre os algoritmos em comparação, em que a medida de inviabilidade da melhor solução h_{otm} é inferior a 10^{-8} . Se estivermos interessados na avaliação dos algoritmos tendo em conta o valor da mediana (ou da média) entre as 30 rodadas, então f_{med} (ou f_{medio}), a mediana (ou a média) dos valores da função encontrados pelo algoritmo s no problema p , deve ser usado em (5.2) em vez de

f. Neste caso, deve também ser usada a mediana da medida de inviabilidade $h_{med} \leq 10^{-8}$ para considerar que o ponto é viável. Caso s não resolva o problema p , $n_{f_{medio}}$ toma o valor infinito.

Para um algoritmo $s \in \mathcal{S}$, a comparação do seu desempenho na resolução de um problema $p \in \mathcal{P}$, em relação ao desempenho do melhor algoritmo, é feita utilizando-se a razão de desempenho:

$$r_{p,s} = \frac{m_{p,s}}{\min\{m_{p,s} : \forall s \in \mathcal{S}\}}. \quad (5.3)$$

A razão de desempenho mostra o comportamento de um algoritmo na resolução de um determinado problema. Para uma avaliação geral do desempenho do algoritmo, é definido o perfil de desempenho de um algoritmo, dado por:

$$\rho_s(\tau) = \frac{1}{n_p} \text{card}\{p \in \mathcal{P} : r_{p,s} \leq \tau\},$$

onde n_p é o número de problemas no conjunto \mathcal{P} . A cardinalidade (card) representa o número de problemas no conjunto, para os quais a razão de desempenho $r_{p,s}$ é menor ou igual a τ para o algoritmo s . A função $\rho_s(\tau)$ é a probabilidade (para o algoritmo $s \in \mathcal{S}$) da razão de desempenho $r_{p,s}$ tomar um valor inferior ou igual ao fator $\tau \in \mathbb{R}$ da melhor proporção possível. Ou seja, $\rho_s(\tau)$ é a função de distribuição para a razão de desempenho.

O valor de $\rho_s(1)$ corresponde à proporção de problemas para os quais o algoritmo s obteve melhor desempenho comparativamente aos outros algoritmos, ou seja, é a probabilidade de que o algoritmo ganhe dos demais. Quanto maior for o valor de ρ_s , melhor é o desempenho do algoritmo s . Por outro lado, para valores elevados de τ , o $\rho_s(\tau)$ mede a robustez do algoritmo. O algoritmo com o maior valor de $\rho_s(\tau)$ é aquele que resolve mais problemas no conjunto \mathcal{P} . É de salientar que, $r_{p,s}$ toma o valor infinito se o problema p não é resolvido pelo algoritmo s pelo critério dado em (5.2).

5.2.1 Experimento para os problemas descritos em [3]

Nesta seção apresentamos e analisamos os resultados numéricos dos experimentos realizados com os Algoritmos FDDS e FDDSRBF para resolver o conjunto de problemas teste descrito

em [3]. Neste experimento os resultados são obtidos após 30 rodadas independentes dos Algoritmos FDDS e FDDSRBF, e comparados com os Algoritmos ALtype 3, EPGO e P-BF AFS, apresentados em [18, 10, 46], respectivamente.

A Figura 5.1 mostra o perfil de desempenho do número médio de avaliações da função objetivo para os Algoritmos FDDS, FDDSRBF, P-BF AFS e EPGO, ao resolverem o conjunto de problemas da coleção descrita em [3]. Neste caso, para avaliar se o algoritmo resolveu determinado problema, utilizamos os resultados da melhor rodada realizada no experimento. Podemos observar na figura que os Algoritmos FDDSRBF e FDDS são mais robustos que os

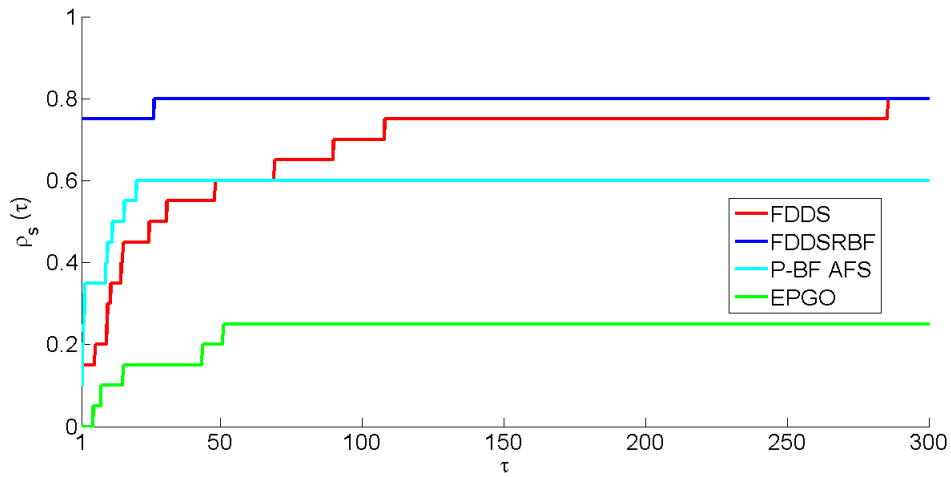


Figura 5.1: Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da melhor rodada para os problemas descritos em [3]

demaís, resolvendo 80% dos problemas da coleção. Além disso, o FDDSRBF é o algoritmo mais eficiente pois resolve 75% dos problemas gastando o menor número médio de avaliações da função objetivo $n_{f_{medio}}$, enquanto que o FDDS gasta menos avaliações em apenas 15% dos problemas. O P-BF AFS resolveu 60% dos problemas, sendo mais eficiente apenas em 10% deles. O Algoritmo EPGO não conseguiu resolver problemas com menor número de avaliações de função em relação aos demais algoritmos. Além disso, ele apresentou a menor robustez dentre os algoritmos analisados.

A Figura 5.2 mostra o perfil de desempenho, relacionado com o número médio de

avaliações da função objetivo dos Algoritmos FDDS e FDDSRBF, onde utilizamos a mediana dos valores obtidos pelos algoritmos sobre as 30 rodadas independentes. Note-se que neste perfil apenas estão em comparação os algoritmos FDDS e FDDSRBF, visto que para o P-BF AFS não estão disponíveis os valores da mediana da medida de inviabilidade e no EPGO, por não ser um algoritmo estocástico, foi executado apenas uma rodada. Como esperado, o

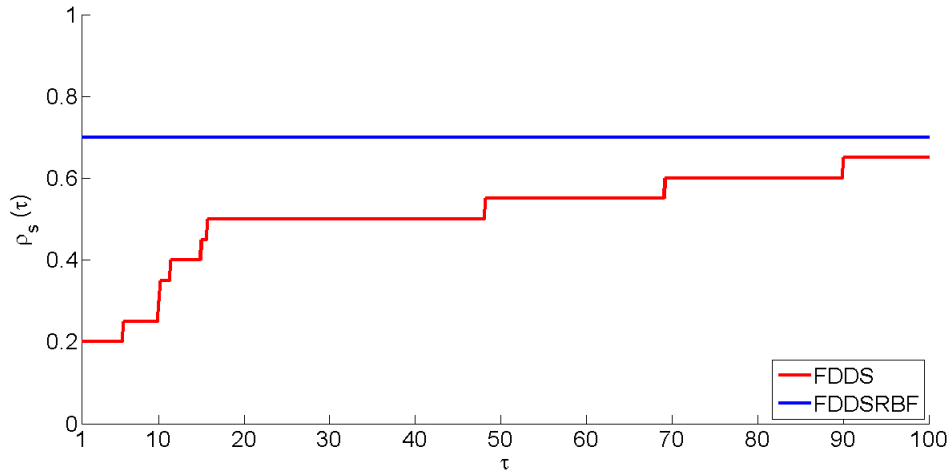


Figura 5.2: Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da mediana das 30 rodadas para os problemas descritos em [3]

FDDSRBF apresentou um desempenho inferior ao obtido por ele na Figura 5.1. No entanto ele continua sendo o algoritmo mais robusto e eficiente, resolvendo 70% dos problemas gastando o menor $n_{f_{medio}}$. Neste caso, o FDDS resolveu 65% dos problemas da coleção, apresentando uma eficiência de apenas 20%.

A Tabela 5.1 apresenta os resultados numéricos desse experimento comparados com o P-BF AFS, EPGO e ALtype 3. Para o ALtype 3, estão disponíveis apenas os resultados para 7 problemas da coleção. As três primeiras colunas da tabela mostram os dados do problema, com a identificação (**P**), o número de variáveis de decisão (n), e o valor ótimo conhecido (f^*). As demais colunas apresentam os resultados obtidos ao longo das 30 rodadas. São eles, a melhor solução obtida pelo algoritmo (f_{otm}), a mediana das soluções obtidas (f_{med}), a medida de inviabilidade da melhor solução (h_{otm}), a mediana da medida

de inviabilidade (h_{med}) e o número médio de avaliações da função objetivo ($n_{f_{medio}}$).

Tabela 5.1: Resultados numéricos para os problemas de [3]

P	n	f^*	Algoritmo	f_{otm}	f_{med}	h_{otm}	h_{med}	$n_{f_{medio}}$
1	5	2.9313e-02	FDDS	0.3730	154.7137	0.00e00	0.00e00	24987
			FDDSRBF	0.2724	141.6525	0.00e00	0.00e00	1000
			P-BF AFS	0.0956	1.4665	7.84e-07	*	6945
			EPGO	0.0625		2.35e-07		39575
2a	9	-4.0000e02	FDDS	-396.099	-310.16	0.00e00	0.00e00	24987
			FDDSRBF	-388.622	0.000	0.00e00	0.00e00	1000
			P-BF AFS	-358.650	-308.664	0.00e00	*	7068
			EPGO	-134.113		8.43e-04		115107
2b	9	-6.0000e02	FDDS	-473.559	-357.508	0.00e00	0.00e00	24987
			FDDSRBF	-284.276	986.439	0.00e00	0.00e00	1000
			P-BF AFS	-378.317	-274.472	0.00e00	*	6963
			EPGO	-768.457		5.30e-04		120057
2c	9	-7.5000e02	FDDS	-749.981	-713.859	0.00e00	0.00e00	24987
			FDDSRBF	-684.722	-544.883	0.00e00	0.00e00	1000
			P-BF AFS	-697.452	-657.349	0.00e00	*	7189
			EPGO	-82.977		8.43e-04		102015
2d	10	-4.0000e02	FDDS	-399.998	-313.613	0.00e00	0.00e00	24987
			FDDSRBF	-400.000	-50.386	0.00e00	0.00e00	1000
			P-BF AFS	-399.118	-394.563	0.00e00	*	6526
			EPGO	-385.170		0.00e00		229773
3a	6	-3.8880e-01	FDDS	-0.3887	-0.3778	0.00e00	0.00e00	29224
			FDDSRBF	-0.3889	-0.3795	0.00e00	0.00e00	940
			P-BF AFS	-0.3888	-0.3842	5.22e-04	*	7495
			EPGO	-0.3861		1.02e-06		48647
3b	2	-3.8881e-01	FDDS	-0.3888	-0.3885	0.00e00	0.00e00	7660
			FDDSRBF	-0.3888	-0.3885	0.00e00	0.00e00	677
			P-BF AFS	-0.3888	-0.3888	0.00e00	*	1041
			EPGO	-0.3888		0.00e00		3449
			ALtype 3	-0.3888		2.90e-05		
4	2	-6.6666e00	FDDS	-6.6666	-6.6666	0.00e00	0.00e00	3391
			FDDSRBF	-6.6666	-6.6665	0.00e00	0.00e00	227
			P-BF AFS	-6.6667	-6.6665	0.00e00	*	493
			EPGO	-6.6666		0.00e00		3547
			ALtype 3	-6.6667		0.00e00		
5	3	2.0116e02	FDDS	201.159	201.159	4.25e-05	1.08e-03	9996

Continua na próxima página

Tabela 5.1 (*Continuação da página anterior*)

P	n	f^*	Algoritmo	f_{otm}	f_{med}	h_{otm}	h_{med}	$n_{f_{medio}}$
6	2	3.7629e02	FDDSRBF	201.159	201.159	0.00e00	0.00e00	1000
			P-BF AFS	201.159	201.159	8.11e-07	*	2999
			EPGO	201.159		1.66e-04		14087
			FDDS	376.293	376.306	0.00e00	0.00e00	9996
			FDDSRBF	376.292	376.293	0.00e00	0.00e00	1000
7	2	-2.8284e00	P-BF AFS	376.293	376.304	0.00e00	*	1335
			EPGO	0.4701		2.05e-05		1523
			ALtype 3	376.292		0.00e00		
			FDDS	-2.8284	-2.8279	0.00e00	0.00e00	8099
			FDDSRBF	-2.8284	-2.8283	0.00e00	0.00e00	90
8	2	-1.1870e02	P-BF AFS	-2.8284	-2.8283	0.00e00	*	920
			EPGO	-2.8058		0.00e00		13187
			ALtype 3	-2.8284		6.64e-07		
			FDDS	-118.705	-86.422	0.00e00	0.00e00	9812
			FDDSRBF	-118.705	-86.422	0.00e00	0.00e00	971
9	6	-1.3402e01	P-BF AFS	-118.704	-118.698	0.00e00	*	1521
			EPGO	-118.704		0.00e00		7621
			ALtype 3	-118.705		2.05e-05		
			FDDS	-13.4019	-13.4015	0.00e00	0.00e00	14993
			FDDSRBF	-13.4019	-13.4019	0.00e00	0.00e00	960
10	2	7.4178e-01	P-BF AFS	-13.4018	-13.4007	0.00e00	*	1839
			EPGO	-13.4026		1.35e-04		68177
			FDDS	0.7418	0.7419	0.00e00	0.00e00	6359
			FDDSRBF	0.7418	0.7419	0.00e00	0.00e00	132
			P-BF AFS	0.7418	0.7418	0.00e00	*	2126
11	2	-5.0000e-01	EPGO	0.7420		0.00e00		6739
			ALtype 3	0.7418		3.08e-06		
			FDDS	-0.5000	-0.4986	0.00e00	0.00e00	8856
			FDDSRBF	-0.5000	-0.4999	0.00e00	0.00e00	82
			P-BF AFS	-0.5000	-0.5000	0.00e00	*	782
12	2	-1.6739e01	EPGO	-0.5000		0.00e00		3579
			ALtype 3	-0.5000		2.45e-07		
			FDDS	-16.7345	-16.2574	0.00e00	0.00e00	9996
			FDDSRBF	-16.7393	-16.7321	0.00e00	0.00e00	931
			P-BF AFS	-16.7389	-16.7389	0.00e00	*	35
13	3	1.8935e02	EPGO	-16.7389		5.36e-06		3499
			FDDS	289.717	289.571	0.00e00	3.86e-03	9996
			FDDSRBF	224.888	289.083	0.00e00	4.59e-04	1000

Continua na próxima página

Tabela 5.1 (Continuação da página anterior)

P	n	f^*	Algoritmo	f_{otm}	f_{med}	h_{otm}	h_{med}	$n_{f_{medio}}$
			P-BF AFS	189.345	253.937	0.00e00	*	4031
			EPGO	195.955		9.21e-04		8085
14	4	-4.5142e00	FDDS	-4.5142	-4.5140	0.00e00	0.00e00	11829
			FDDSRBF	-4.5142	-4.5142	0.00e00	0.00e00	171
			P-BF AFS	-4.5142	-4.5139	0.00e00	*	2028
			EPGO	-4.3460		9.22e-05		19685
15	3	0.0000e00	FDDS	0.0000	0.0000	3.69e-05	1.16e-02	14993
			FDDSRBF	0.0000	0.0000	0.00e00	1.25e-02	969
			P-BF AFS	0.0000	0.0000	9.11e-07	*	3593
			EPGO	0.0000		4.94e-05		1645
16	5	7.0492e-01	FDDS	0.7049	0.7050	0.00e00	0.00e00	128
			FDDSRBF	0.7049	0.7050	0.00e00	0.00e00	22
			P-BF AFS	0.7049	0.7049	0.00e00	*	447
			EPGO	0.7181		2.00e-04		22593

*informação não disponível na fonte

A partir dos resultados apresentados na Tabela 5.1, podemos concluir que nesse experimento os algoritmos propostos apresentam resultados bastante competitivos quando comparados com os Algoritmos P-BF AFS e o EPGO. O Algoritmo FDDS é capaz de atingir a solução com uma boa precisão, exceto para os Problemas 1, 2b e 13. Assim, o FDDS atinge a solução f^* em 17 dos 20 problemas, e os Algoritmos P-BF AFS e EPGO atingem a melhor solução em 16 e 13 problemas, respectivamente. No entanto, o FDDS utiliza um número elevado de avaliações da função objetivo. Além disso, o EPGO também é computacionalmente caro, devido ao alto número de avaliações de função gasto, e apresenta uma qualidade da solução inferior aos demais algoritmos. Por outro lado, o Algoritmo FDDSRBF é bem mais eficiente que os demais algoritmos, de acordo com os resultados apresentados, requerendo um número de avaliações da função objetivo bem inferior aos demais. Portanto, o FDDSRBF mantém uma boa qualidade da solução num custo computacional bastante inferior. Os resultados do ALtype 3 foram apresentados em [18] apenas para os problemas 3b, 4, 6, 7, 8, 10 e 11. Como podemos observar, o ALtype 3 atingiu os valores ótimos f^* nos 7

problemas da coleção, embora algumas soluções tenham inviabilidade superior a ε_h , visto que levaram em consideração uma tolerância da medida de inviabilidade de 10^{-4} .

5.2.2 Experimento para os problemas descritos em [20]

Agora vamos analisar o desempenho dos algoritmos propostos para resolver o conjunto de problemas descritos em [20], obtidos após 30 rodadas independentes. Esses resultados são comparados com os resultados do Algoritmo P-BF AFS para essa mesma coleção de problemas, apresentados em [46] e com análise dos resultados feita sobre a média das 30 rodadas. Por esse motivo, na comparação dos resultados desta coleção utilizamos a média em vez da mediana.

A Figura 5.3 apresenta o perfil de desempenho do número médio de avaliações da função objetivo para os Algoritmos FDDS, FDDSRBF e P-BF AFS, ao resolverem o conjunto de problemas da coleção, considerando que um algoritmo resolve um determinado problema tendo em conta os valores da melhor rodada realizada neste experimento. Como podemos

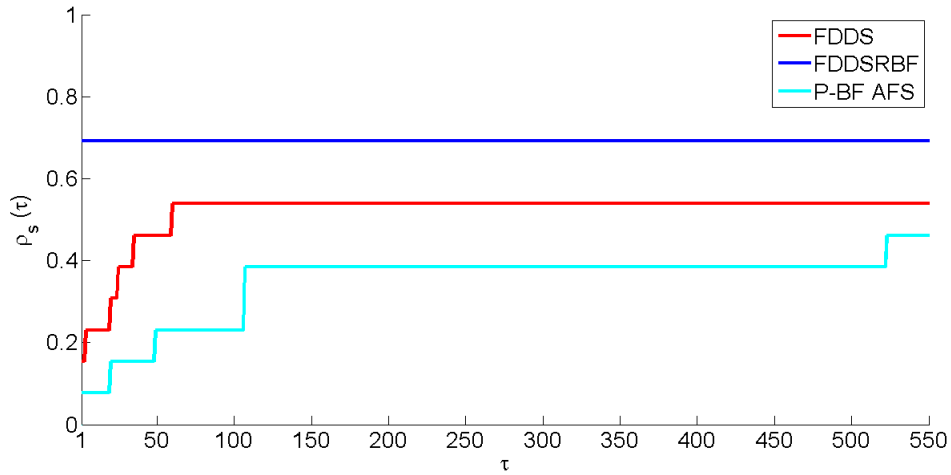


Figura 5.3: Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da melhor rodada para os problemas descritos em [20]

observar na figura, o FDDSRBF apresentou um desempenho razoável, sendo o código mais robusto e eficiente ao resolver 69% dos problemas dessa coleção com o menor $n_{f_{medio}}$.

Seguido dos Algoritmos FDDS e P-BF AFS que resolveram, respectivamente, 54% e 46% dos problemas da coleção.

A Figura 5.4 apresenta os perfis de desempenho dos Algoritmos FDDS e FDDSRBF, relacionado com o número médio de avaliações da função objetivo, considerando que um algoritmo resolve um determinado problema tendo em conta a média dos valores obtidos sobre as 30 rodadas independentes. Neste perfil o P-BF AFS não está em comparação, uma vez que os seus valores médios da medida de inviabilidade não estão disponíveis.

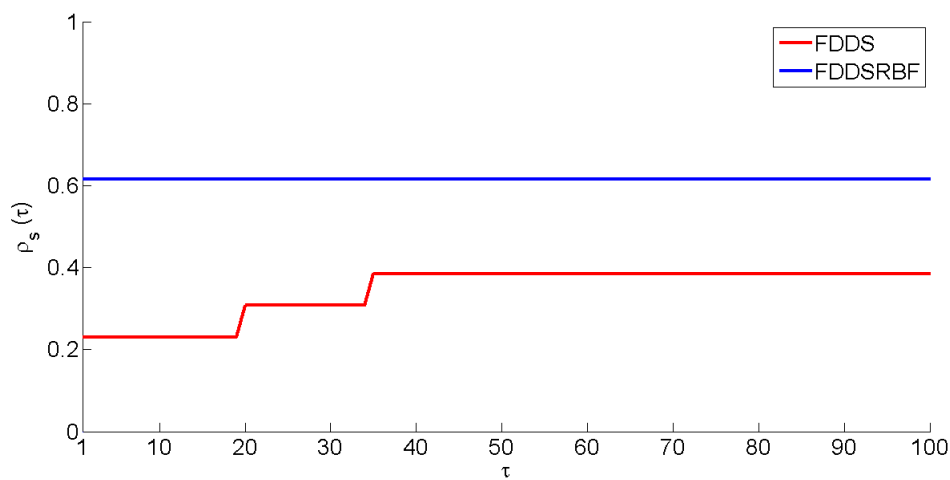


Figura 5.4: Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da média das 30 rodadas para os problemas descritos em [20]

Na Figura 5.4, podemos observar que o desempenho do FDDSRBF foi próximo do observado na melhor rodada, resolvendo 61% dos problemas com o menor número médio de avaliações da função objetivo. O FDDS apresentou um desempenho inferior ao da melhor rodada, resolvendo apenas 38% dos problemas e gastando menos avaliações da função objetivo em apenas 23% deles.

A Tabela 5.2 mostra os resultados deste experimento numérico ao resolver o conjunto de problemas, descritos em [20], obtidos após 30 rodadas independentes. As colunas 5 e 7 apresentam os valores médios da função objetivo (f_{medio}) e da medida de inviabilidade (h_{medio}), tomados sobre as 30 rodadas. Neste caso, utilizamos os valores médios para

comparar com os valores médios apresentados em [46].

Tabela 5.2: Resultados numéricos para os problemas de [20]

P	n	f^*	Algoritmo	f_{otm}	f_{medio}	h_{otm}	h_{medio}	$n_{f_{medio}}$
G1	13	-15.00000	FDDS	-14.99997	-53.59259	0.00e00	1.57e01	59128
			FDDSRBF	-14.99963	-44.01687	0.00e00	1.19e01	1000
			P-BF AFS	-14.99999	-14.99992	0.00e00	*	48929
G2	20	-0.803619	FDDS	-0.797456	-0.759612	0.00e00	0.00e00	99942
			FDDSRBF	-0.614015	-0.355211	0.00e00	0.00e00	1000
			P-BF AFS	-0.764816	-0.730774	0.00e00	*	104312
G3	10	-1.000000	FDDS	-0.213367	-0.023857	0.00e00	0.00e00	49972
			FDDSRBF	-0.998023	-0.807604	0.00e00	0.00e00	1000
			P-BF AFS	-1.000008	-0.999575	8.32e-06	*	51994
G4	5	-30665.539	FDDS	-30665.400	-30529.656	0.00e00	0.00e00	24987
			FDDSRBF	-30663.020	-30326.631	0.00e00	0.00e00	1000
			P-BF AFS	-30665.538	-30665.524	9.84e-05	*	102188
G5	4	5126.4981	FDDS	5327.1540	4295.8009	8.32e-03	2.32e-01	19990
			FDDSRBF	5126.5570	4407.6813	1.83e-06	2.15e-01	1000
			P-BF AFS	5126.4983	5128.4768	1.15e-05	*	112853
G6	2	-6961.8139	FDDS	-6933.4960	-6809.6122	0.00e00	0.00e00	9996
			FDDSRBF	-6958.4010	-6948.5690	0.00e00	0.00e00	1000
			P-BF AFS	-6961.8138	-6961.8127	0.00e00	*	106718
G7	10	24.306209	FDDS	24.48737	26.65462	0.00e00	0.00e00	49972
			FDDSRBF	24.33933	25.1571	0.00e00	6.69e-10	1000
			P-BF AFS	24.63254	25.43844	0.00e00	*	117449
G8	2	-0.095825	FDDS	-0.095820	-0.091223	0.00e00	0.00e00	1414
			FDDSRBF	-0.095822	-0.067832	0.00e00	0.00e00	462
			P-BF AFS	-0.095825	-0.095824	0.00e00	*	8967
G9	7	680.63006	FDDS	680.78720	681.18732	0.00e00	0.00e00	34981
			FDDSRBF	680.65960	680.8659	0.00e00	7.61e-10	1000
			P-BF AFS	680.64912	680.66745	0.00e00	*	106406
G10	8	7049.3307	FDDS	7444.0250	11153.4672	0.00e00	1.05e-04	39978

Continua na próxima página

Tabela 5.2 (Continuação da página anterior)

P	n	f^*	Algoritmo	f_{otm}	f_{medio}	h_{otm}	h_{medio}	$n_{f_{medio}}$
G11	2	0.750000	FDDSRBF	7271.7960	9526.0102	0.00e00	1.64e-09	1000
			P-BF AFS	7077.5240	7198.3822	0.00e00	*	125880
			FDDS	0.750976	0.841766	0.00e00	0.00e00	9996
G12	3	-1.000000	FDDSRBF	0.749954	0.750277	0.00e00	0.00e00	493
			P-BF AFS	0.749900	0.749901	1.00e-04	*	75997
			FDDS	-0.999991	-0.999942	0.00e00	0.00e00	430
G13	5	0.0539498	FDDSRBF	-0.999998	-0.999951	0.00e00	0.00e00	22
			P-BF AFS	-1.000000	-0.999998	0.00e00	*	11494
			FDDS	0.061505	1.501963	0.00e00	2.05e-05	24987
			FDDSRBF	0.078153	1.305580	0.00e00	1.40e-10	1000
			P-BF AFS	0.056265	0.289244	1.52e-03	*	95508

*informação não disponível na fonte

Podemos observar na Tabela 5.2 que, apesar de apresentar um número bastante elevado de avaliações da função objetivo, o P-BF AFS obteve melhores aproximações à solução ótima conhecida, f^* , para a maioria dos problemas da coleção, embora tenha apresentado inviabilidade em 5 dos 13 problemas. O FDDS apresentou o pior desempenho relativamente à aproximação da solução ótima conhecida, f^* . Por outro lado, o FDDSRBF conseguiu manter um bom nível de aproximação de f^* , exceto para os problemas G2, G10 e G13, apresentando, porém, um gasto com o número médio de avaliações da função objetivo bastante competitivo relativamente aos demais algoritmos analisados neste experimento.

5.2.3 Experimento para os problemas descritos em [47]

Nesta seção, mostramos a eficácia dos algoritmos propostos ao resolver problemas de aplicações reais, que compreende um conjunto de nove problemas de desenho de engenharia, apresentados em [47]. Os resultados, obtidos após 30 rodadas independentes, foram

comparados com os resultados do Algoritmo A2-SF, apresentado em [30], para essa mesma coleção de problemas.

A Figura 5.5 apresenta o perfil de desempenho do número de avaliações da função objetivo para os Algoritmos FDDS, FDDSRBF e A2-SF, ao resolverem a coleção de problemas descrita em [47], considerando que um algoritmo resolve um determinado problema tendo em conta os resultados da melhor rodada realizada no experimento. Como observado na

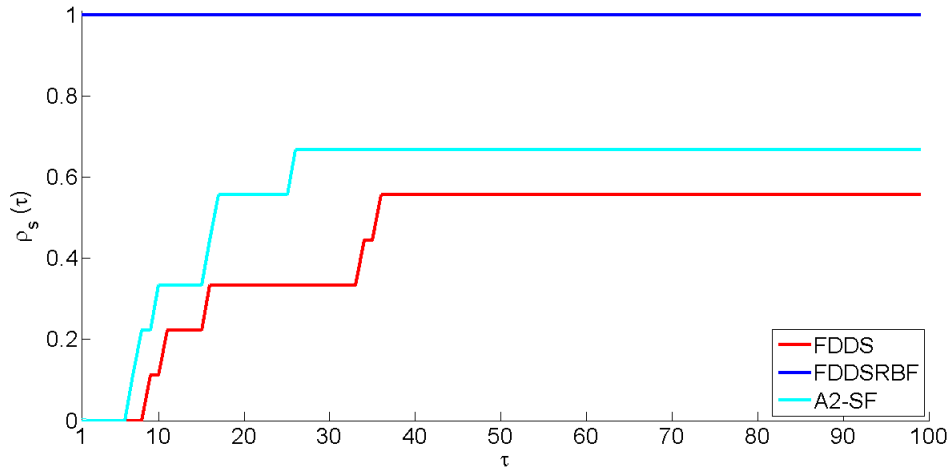


Figura 5.5: Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da melhor rodada para os problemas descritos em [47]

figura, o FDDSRBF apresentou um desempenho muito bom em relação aos demais algoritmos, resolvendo 100% dos problemas da coleção com o menor valor de $n_{f_{medio}}$. O A2-SF foi o segundo mais robusto resolvendo 66% dos problemas, e o FDDS resolveu 55% dos problemas da coleção.

A Figura 5.6 apresenta os perfil de desempenho do número médio de avaliações da função objetivo para os Algoritmos FDDS, FDDSRBF e A2-SF, considerando que um algoritmo resolve um determinado problema tendo em conta a mediana dos valores obtidos sobre as 30 rodadas independentes. Como podemos observar na figura, o FDDSRBF é o algoritmo mais robusto e eficiente, resolvendo 77%. O Algoritmo A2-SF resolveu 55% dos problemas da coleção, gastando o menor $n_{f_{medio}}$ em apenas 22% dos problemas. Enquanto que o FDDS

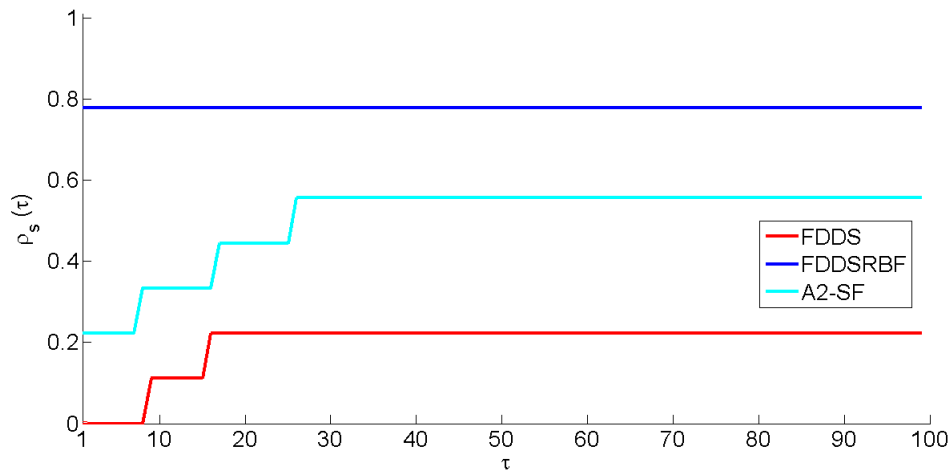


Figura 5.6: Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da mediana das 30 rodadas para os problemas descritos em [47]

apresentou a pior eficiência e robustez na comparação.

A Tabela 5.3 apresenta os resultados numéricos desse experimento para as 30 rodadas independentes dos Algoritmos FDDS, FDDSRBF e A2-SF.

Tabela 5.3: Resultados numéricos para os problemas de engenharia descritos em [47]

P	n	f^*	Algoritmo	f_{otm}	f_{med}	h_{otm}	h_{med}	$n_{f_{medio}}$
Cyl. Vessel	5	5868.7650	FDDS	5996.0260	6671.9015	0.00e00	0.00e00	19990
			FDDSRBF	5885.4030	6263.9640	0.00e00	0.00e00	1000
			A2-SF	5898.3626	6327.0383	0.00e00	0.00e00	10966
Disc Brake	4	0.1274	FDDS	0.1274	0.1456	0.00e00	0.00e00	14971
			FDDSRBF	0.1274	0.1274	0.00e00	0.00e00	420
			A2-SF	0.1274	0.1283	0.00e00	0.00e00	6770
Four Bar Truss	4	1400.000	FDDS	1400.000	1400.000	0.00e00	0.00e00	106
			FDDSRBF	1400.000	1400.000	0.00e00	0.00e00	13
			A2-SF	1400.000	1400.000	0.00e00	0.00e00	336
Heat Exch.	8	7049.2480	FDDS	7095.1140	9527.4565	0.00e00	0.00e00	39978
			FDDSRBF	7051.1860	7666.1930	0.00e00	0.00e00	1000

Continua na próxima página

Tabela 5.3 (Continuação da página anterior)

P	n	f^*	Algoritmo	f_{otm}	f_{med}	h_{otm}	h_{med}	$n_{f_{medio}}$
			A2-SF	7075.0293	8340.3915	0.00e00	0.00e00	17826
Speed	7	2994.4991	FDDS	2994.4840	3132.9250	0.00e00	0.00e00	33586
Reducer			FDDSRBF	2994.4840	3089.5885	0.00e00	3.89e-09	1000
			A2-SF	2994.4840	2994.5185	0.00e00	0.00e00	15146
Tubular	2	26.5313	FDDS	27.1831	28.6203	0.00e00	0.00e00	9996
Column			FDDSRBF	26.5314	27.0672	0.00e00	0.00e00	969
			A2-SF	26.5342	26.6429	0.00e00	0.00e00	5884
Tension	3	0.0127	FDDS	0.0127	0.0131	0.00e00	0.00e00	12102
Spring			FDDSRBF	0.0127	0.0129	0.00e00	0.00e00	774
			A2-SF	0.0127	0.0140	0.00e00	0.00e00	7598
Three	2	263.8958	FDDS	263.8980	264.8168	0.00e00	0.00e00	9996
Bar			FDDSRBF	263.8959	263.8964	0.00e00	0.00e00	910
Truss			A2-SF	263.9086	264.0120	0.00e00	0.00e00	6649
Welded	4	2.3809	FDDS	2.5579	3.9276	0.00e00	0.00e00	19990
Beam			FDDSRBF	2.4568	3.2971	0.00e00	0.00e00	1000
			A2-SF	2.5942	5.2176	0.00e00	0.00e00	11016

*informação não disponível na fonte

A partir dos resultados apresentados na Tabela 5.3 para o conjunto de problemas de engenharia, podemos concluir que os nossos algoritmos são capazes de atingir a solução com uma boa precisão, exceto para os Problemas Cylindrical Vessel, Heat Exchange e Welded Beam.

5.3 Análise do desempenho dos algoritmos

O intuito, nesta seção, é analisar de uma maneira geral o comportamento dos algoritmos propostos relativamente ao número médio de avaliações da função objetivo gastos e ao número de problemas resolvidos, de acordo com (5.2), para os três experimentos numéricos realizados. De uma maneira geral, podemos observar nas Tabelas 5.1, 5.2 e 5.3, que os Al-

goritmos FDDS e FDDSRBF apresentam soluções viáveis em praticamente todos os problemas das três coleções.

A Figura 5.7 apresenta o perfil de desempenho dos algoritmos propostos relativamente ao número médio de avaliações da função objetivo, para a melhor das 30 rodadas realizadas ao resolver os 42 problemas das três coleções utilizadas nos experimentos.

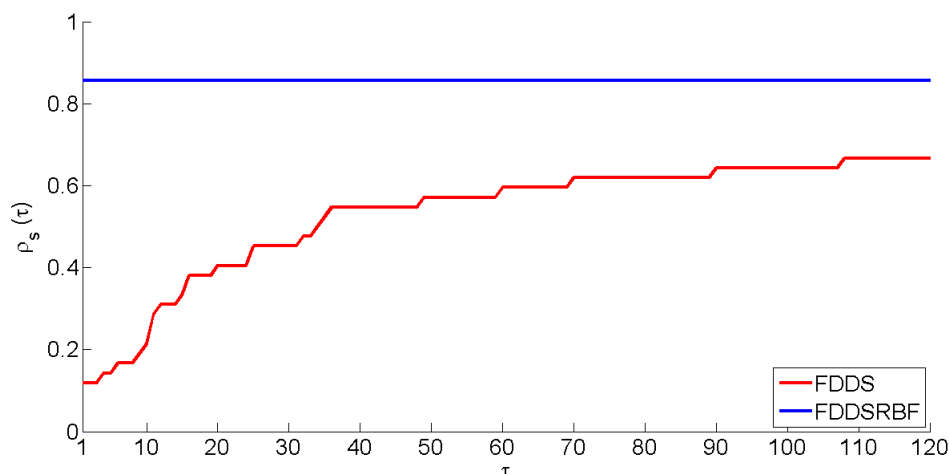


Figura 5.7: Perfil de desempenho do $n_{f_{medio}}$, baseado nos valores da melhor rodada para as três coleções de problemas teste

De acordo com a Figura 5.7, podemos concluir que o Algoritmo FDDSRBF é mais robusto e eficiente do que o FDDS. A eficiência do FDDSRBF já era esperada visto que seu objetivo principal era a economia no número de avaliações da função objetivo. É ainda de realçar que, o FDDSRBF surpreendeu ao apresentar robustez superior ao FDDS, uma vez que ele utiliza aproximações da função objetivo na seleção do melhor ponto tentativo. Provavelmente isso se deve ao fato do número de pontos tentativos gerados, em cada iteração, no FDDSRBF ser bem superior ao gerado no FDDS.

Capítulo 6

Conclusões e trabalhos futuros

O objetivo desta tese foi o desenvolvimento de novos algoritmos estocásticos de otimização global, eficientes e robustos para resolver problemas de otimização não lineares com restrições, possivelmente não convexos e não suaves, em que o custo computacional de avaliações da função objetivo é alto.

Devido ao caráter estocástico dos algoritmos, foi necessário um estudo acerca da teoria de medida em espaços de probabilidade, variáveis e vetores aleatórios, função de distribuição de probabilidade, em particular a distribuição normal multivariada, convergência no contexto probabilístico e aspectos gerais de algoritmos de busca estocástica.

Seguindo as ideias de [38], foi feita a análise da convergência em probabilidade de um algoritmo estocástico de busca direta bastante geral para problemas restritos a conjuntos viáveis fáceis, no sentido de que é fácil gerar iterandos viáveis. Discutimos três algoritmos estocásticos existentes na literatura que se encaixam na estrutura do algoritmo geral. Em seguida estudamos a convergência em probabilidade do algoritmo LRS, que é um caso particular do algoritmo geral analisado, onde os iterandos são gerados perturbando-se todas as componentes do melhor ponto corrente por um vetor aleatório com distribuição normal multivariada. Discutimos também uma maneira de economizar avaliações de função usando os chamados modelos de superfície de resposta. Nossa contribuição no contexto de problemas com restrições de limites simples na variáveis foi o estudo da convergência em

probabilidade do Algoritmo DDS, com base nas ideias de [38] e no Lema de Borel-Cantelli.

Uma vez entendida as condições para a convergência de métodos estocásticos de busca direta viáveis, estendemos nosso estudo para problemas com restrições mais gerais, principal foco de interesse desta tese. Neste caso os algoritmos devem de algum modo controlar a inviabilidade, o que fizemos através da estratégia dos métodos de filtro que utiliza valores da função objetivo e de uma medida de inviabilidade. Propusemos um algoritmo geral estocástico de filtro, onde os iterandos são vetores aleatórios cujas realizações são geradas de acordo com alguma distribuição de probabilidade. O algoritmo é bastante geral no sentido de que a sua convergência quase certamente é garantida independentemente de qual distribuição de probabilidade tenha sido utilizada no cálculo dos iterandos, desde que algumas hipóteses sejam satisfeitas. Assim, uma análise de condições suficientes para que o algoritmo proposto convirja quase certamente foi apresentada.

Em seguida, apresentamos o Algoritmo FDDS que se encaixa na estrutura do algoritmo geral, em que os iterandos são calculados adicionando-se perturbações aleatórias com distribuição normal nas coordenadas, escolhidas de forma dinâmica, do melhor ponto corrente. No entanto, com a estratégia de gerar múltiplos pontos tentativos em cada iteração, o custo computacional associado ao número de avaliações da função objetivo pode ser bastante elevado. Com o intuito de ultrapassar esta desvantagem, usamos um modelo cúbico de funções de base radial para aproximar a função objetivo, dando origem ao Algoritmo FDDSRBF. Para a escolha do melhor ponto tentativo, recorreu-se também a uma metodologia de filtro, e definiu-se um novo filtro que utiliza valores da medida de inviabilidade e do modelo cúbico de aproximação da função objetivo.

Foram apresentados resultados computacionais promissores, comparando-se o desempenho dos algoritmos propostos com alguns algoritmos existentes na literatura para 42 problemas de três conjuntos diferentes. O Algoritmo FDDSRBF mostrou-se bastante eficiente e robusto, relativamente aos algoritmos utilizados na comparação, com uma significativa redução do número de avaliações de função.

Uma dificuldade apresentada pelo Algoritmo FDDSRBF, para alguns problemas das coleções utilizadas, foi manter a mediana das soluções obtidas próximo da melhor solução. Neste

sentido sugerimos, como um tópico futuro, um estudo com o intuito de melhorar a consistência dos resultados numéricos obtidos pelo **FDDSRBF**. Adicionalmente, sugerimos fazer uma pesquisa sobre outros métodos estocásticos de busca direta que se enquadram nos moldes do Algoritmo geral estocástico de filtro e, assim, analisar a convergência quase certamente de tais métodos. Outra direção de pesquisa futura é estudar métodos estocásticos para os quais existem parâmetros aleatórios na formulação do problema, por exemplo.

Em suma, as principais contribuições desta tese residem na apresentação de algoritmos estocásticos em que o controle da inviabilidade é feito através da estratégia dos métodos de filtro, com resultados teóricos de convergência quase certamente e experimentos computacionais.

Referências Bibliográficas

- [1] M. M. Ali e W. X. Zhu. A penalty function-based differential evolution algorithm for constrained global optimization. *Comput. Optim. Appl.*, 54(1):707–739, 2013.
- [2] M. Bazaraa, H. Sherali, e C. Shetty. *Nonlinear Programming Theory and Algorithms*. Athena Scientific, Wiley, New York, 2nd edition, 2006.
- [3] E. G. Birgin, C. A. Floudas, e J. M. Martínez. Global minimization using an augmented Lagrangian method with variable lower-level constraints. *Math. Program.*, 125:139–162, 2010.
- [4] E. G. Birgin e J. M. Gentil. Evaluating bound-constrained minimization software. *Comput. Optim. Appl.*, 53(2):347–373, 2012.
- [5] E. G. Birgin e J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *J. Comput. Appl. Math.*, 23:101–125, 2002.
- [6] C. M. Chin. A local convergence theory of a filter line search method for nonlinear programming. Technical report, Numerical Optimization Report, Department of Statistics, University of Oxford, 2003.
- [7] C. M. Chin e R. Fletcher. On the global convergence of an SLP-filter algorithm that takes EQP steps. *Math. Program.*, 96(1):161–177, 2003.
- [8] P. D. Conejo, E. W. Karas, L. G. Pedroso, A. A. Ribeiro, e M. Sachine. Global convergence of trust-region algorithms for convex constrained minimization without derivatives. *Appl. Math. Comput.*, 220:324–330, 2013.

- [9] M. F. P. Costa, F. P. Fernandes, e A. M. A. C. Rocha. Multiple solutions of mixed variable optimization by multistart Hooke and Jeeves filter method. *Applied Mathematical Sciences*, 8:2163–2179, 2014.
- [10] G. Di Pillo, S. Lucidi, e F. Rinaldi. An approach to constrained global optimization based on exact penalty functions. *J. Glob. Optim.*, 54:251–260, 2012.
- [11] E. D. Dolan e J.J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2):201–213, 2009.
- [12] R. Durrett. *Probability: Theory and Examples*. Cambridge University Press, Cambridge, 2013.
- [13] P. S. Ferreira, E. W. Karas, e M. Sachine. A globally convergent trust-region algorithm for unconstrained derivative-free optimization. *Comput. Appl. Math.*, 34:1075–1103, 2015.
- [14] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, e A. Wachter. Global convergence of trust-region SQP-filter algorithm for general nonlinear programming. *SIAM J. Optim.*, 13:635–659, 2002.
- [15] R. Fletcher e S. Leyffer. Nonlinear programming without a penalty function. *Math. Program.*, 91:239–269, 2002.
- [16] R. Fletcher e S. Leyffer. Filter-type algorithms for solving systems of algebraic equations and inequalities. In G. di Pillo e A. Murli, editors, *Advances in Optimization and Numerical Analysis*, pages 259–278. High Performance Algorithms and Software for Nonlinear Optimization, Kluwer, 2003.
- [17] R. Fletcher, S. Leyffer, e Ph. L. Toint. On the global convergence of a filter-SQP algorithm. *SIAM J. Optim.*, 13(1):44–59, 2002.

- [18] M. L. N. Gonçalves, J. G. Melo, e L. F. Prudente. Augmented Lagrangian methods for nonlinear programming with possible infeasibility. *J. Glob. Optim.*, 63:297–318, 2015.
- [19] C. C. Gonzaga, E. W. Karas, e M. Vanti. A globally convergent filter method for nonlinear programming. *SIAM J. Optimiz.*, 14(3):646–669, 2003.
- [20] A. R. Hedar e M. Fukushima. Derivative-free filter simulated annealing method for constrained continuous global optimization. *J. Glob. Optim.*, 35(4):521–549, 2006.
- [21] E. M. T. Hendrix e B. G. Tóth. *Introduction to Nonlinear and Global Optimization, Optimization and its Applications*, volume 37. Springer, Berlin, 2010.
- [22] R. Hooke e T. A. Jeeves. Direct search solution of numerical and statistical problems. *J. Associated Comput.*, 8:212–229, 1961.
- [23] B. R. James. *Probabilidade: um curso em nível intermediário*. Impa, Rio de Janeiro, Brasil, 1a edition, 1981.
- [24] J. S. R. Jang, C. T. Sun, e E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, Upper Saddle River, NJ, 1a edition, 1997.
- [25] P. Kall e S. W. Wallace. *Stochastic Programming*. John Wiley & Sons, Chichester, 1994.
- [26] E. W. Karas, C. C. Gonzaga, e A. A. Ribeiro. Local convergence of filter methods for equality constrained non-linear programming. *Optimization*, 59:1153–1171, 2010.
- [27] E. W. Karas, A. P. Oening, e A. A. Ribeiro. Global convergence of slanting filter methods for nonlinear programming. *Appl. Math. Comput.*, 200:486–500, 2008.
- [28] E. W. Karas, L. G. Pedroso, e A. Verdério. On the construction of quadratic models for derivative-free trust-region algorithms. *EURO Journal on Computational Optimization*, 2017.

- [29] G. Liuzzi, S. Lucidi, e V. Piccialli. Exploiting derivative-free local searches in direct-type algorithms for global optimization. *Comput. Optim. Appl.*, 65:449–475, 2016.
- [30] M. J. F. G. Macêdo, M. F. P. Costa, A. M. A. C. Rocha, e E. W. Karas. *Combining filter method and dynamically dimensioned search for constrained global optimization*, pages 119–134. Springer International Publishing, Cham, 2017.
- [31] J. Matyas. Random optimization. *Automation and Remote Control*, 26:244–251, 1965.
- [32] S. L. Neitsch, J. G. Arnold, J. R. Kiniry, J. R. Williams, e K. W. King. Soil and water assessment tool theoretical documentation - version 2000. Technical report, Texas Water Resources Institute, 2000. TWRI Report TR-191.
- [33] J. Nocedal e S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, 2nd edition, 2006.
- [34] A. I. Pereira, O. Ferreira, S. P. Pinho, e E. M. G. P. Fernandes. *Multilocal programming and applications*, pages 157–186. Springer Berlin Heidelberg, 2013.
- [35] G. A. Perçaro, A. A. Ribeiro, e E. W. Karas. Global convergence of a general filter algorithm based on an efficiency condition of the step. *Appl. Math. Comput.*, 219:9581–9597, 2013.
- [36] M. J. D. Powell. The theory of radial basis function approximation in 1990. In W. Light, editor, *Advances in numerical analysis. Vol. 2. Wavelets, subdivision algorithms and radial basis functions*, pages 105–210. Oxford University Press, 1992.
- [37] C. J. Price, M. Reale, e B. L. Robertson. Stochastic filter methods for generally constrained global optimization. *J. Glob. Optim.*, 65:441–456, 2016.
- [38] R. G. Regis. Convergence guarantees for generalized adaptive stochastic search methods for continuous global optimization. *European Journal of Operational Research*, 207:1187–1202, 2010.

- [39] R. G. Regis. Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. *Computers & Operations Research*, 38:837–853, 2011.
- [40] R. G. Regis. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Eng. Optimiz.*, 46:218–243, 2014.
- [41] R. G. Regis e C. A. Shoemaker. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, 19(4):497–509, 2007.
- [42] R. G. Regis e C. A. Shoemaker. Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Eng. Optimiz.*, 45:529–555, 2013.
- [43] S. I. Resnick. *A Probability Path*. Birkhauser, Boston, 1999.
- [44] A. A. Ribeiro e E. W. Karas. *Otimização Contínua: aspectos teóricos e computacionais*. Cengage Learning, São Paulo, Brazil, 2013. Em português.
- [45] A. A. Ribeiro, E. W. Karas, e C. C. Gonzaga. Global convergence of filter methods for nonlinear programming. *SIAM J. Optim.*, 19(3):1231–1249, 2008.
- [46] A. M. A. C. Rocha, M. F. P. Costa, e E. M. G. P. Fernandes. A filter-based artificial fish swarm algorithm for constrained global optimization: theoretical and practical issues. *J. Global Optim.*, 60:239–263, 2014.
- [47] A. M. A. C. Rocha e E. M. G. P. Fernandes. Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems,. *Int. J. Comput. Math.*, 86:1932–1946, 2009.
- [48] S. M. Ross. *Introduction to Probability Models*. Academic Press, Berkeley, California, 2007.

- [49] A. Shapiro, D. Dentcheva, e A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*. Siam and MPS, Philadelphia, 2009.
- [50] J. C. Spall. *Introduction to stochastic search and optimization: estimation, simulation and control*. Wiley-interscience, New Jersey, 2003.
- [51] B. A. Tolson, M. Asadzadeh, e A. Zecchin. Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization. *Water Resour. Res.*, 45, 2009.
- [52] B. A. Tolson e C. A. Shoemaker. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resour. Res.*, 43, 2007.
- [53] K. Q. Ye, W. Li, e A. Sudjianto. Algorithmic construction of optimal symmetric latin hypercube designs. *Statistical Planning and Inference*, 90(1):145–159, 2000.